Echoes of Movement: Assessing the Viability of Radar Technology in Detecting Gait Anomalies for Healthcare

Bhavith Manapoty

4th Year Project Report
Artificial Intelligence and Computer Science
School of Informatics
University of Edinburgh

2024

Abstract

This project explores the viability of using radar technology to detect anomalies in human movement patterns for healthcare applications. Traditional methods for monitoring human movement patterns include cameras and wearable sensors. However, these methods have limitations, such as privacy concerns and user compliance. Radar technology offers a non-invasive, privacy-preserving alternative capable of accurately capturing detailed movement data. This research, conducted as part of the broader Project Feather at the University of Edinburgh, evaluates the feasibility and effectiveness of radar for continuous health monitoring in an assisted living context. The study collected data from seven subjects performing activities in a simulated environment, showcasing normal and abnormal walking patterns. The collected data was pre-processed and analysed using various computational models, including Non-Negative Matrix Factorization (NMF), Gaussian Mixture Model, One-Class Support Vector Machine, Autoencoders, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) and a combined CNN+LSTM model. The results demonstrated the potential of radar technology in human healthcare applications, showing promising results in identifying gait anomalies. The CNN+LSTM model yielded the best performance among the supervised models, while the NMF model yielded the best among the unsupervised models. This study highlights the potential of radar technology as a viable tool in the domain of assisted living and digital healthcare, offering a non-intrusive and privacy-preserving alternative to existing data collection methods. Future work can explore the potential of this technology in detecting collective and contextual anomalies.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics Committee.
Ethics application number: 671984
Date when approval was obtained: 2023-12-01
The participants' information sheet and a consent form are included in the appendix.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Bhavith Manapoty*)

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Kianoush Nazarpour, for giving me the incredible opportunity to work on this research project and for their invaluable guidance and unwavering support throughout this study. His expertise, patience, and encouragement have been instrumental in shaping this thesis and helping me grow as a researcher.

Secondly, I would like to express my heartfelt gratitude to Saber Mirzaee Bafti for their constant support and feedback in weekly meetings. Their willingness to share their knowledge and provide constructive suggestions has contributed significantly to the success of this project.

I am also incredibly grateful to the entire research team working on the Feather Project for creating a stimulating and collaborative environment that fostered my learning and growth. Their dedication, enthusiasm and diverse perspectives have enriched my research experience and inspired me to push the boundaries of my work.

Finally, I would like to thank my friends and family for their unwavering love, understanding, and encouragement throughout this challenging journey. Their constant support and belief in me have been a source of strength and motivation, enabling me to persevere through the most challenging times and reach this significant milestone.

I extend my heartfelt appreciation to all those mentioned above and the many others who have contributed to my academic journey in countless ways. I am deeply grateful for having you all by my side.

Table of Contents

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Assisted living and digital healthcare have seen significant advancements in recent years, with a growing emphasis on the early detection of health issues [8]. One crucial aspect of this field is the monitoring and analysis of human activities and movement to identify anomalies that may indicate underlying health problems, falls or other emergencies. Popular methods for data collection in this domain include wearable sensors or cameras placed in the environment. However, these approaches come with limitations, such as privacy concerns, the need for constant wear and restricted coverage areas.

To overcome these challenges, radar technology has emerged as a promising alternative for noninvasive, privacy-preserving, and efficient data collection [36]. Radar systems can be passively installed in a fixed location, eliminating the need for individuals to wear sensors continuously. Through sophisticated algorithms, radar technology enables the extraction of valuable information, including location coordinates, velocity, and acceleration of a person. By analyzing this data, patterns and deviations from normal behaviour can be identified, facilitating early detection of potential health issues.

Project Feather is a research project at The University of Edinburgh that aims to facilitate health and well-being by developing systems for early recognition of urinary tract infections (UTI). Project Feather aims to analyse human movement data, activities, and cognitive function through interactions with intelligent agents to detect the presence of a UTI before it gets fatal. As a subset of Project Feather, the current study investigates the potential and viability of using radar for noninvasive data collection of human movement and activities. By leveraging the capabilities of a single radar sensor and developing robust computational models, this project aims to demonstrate the potential of radar technology in detecting anomalies in human movement patterns and its applications in assisted living and digital healthcare.

In summary, this project's motivation stems from the need for a non-intrusive, privacy-preserving, and efficient method for detecting anomalies in human movement patterns. By exploring using a single radar sensor for data collection, this project seeks to advance assisted living and digital healthcare systems, ultimately improving the quality of life for individuals who require continuous monitoring and support. The integration of this project with Project Feather further emphasizes the significance of radar technology in the healthcare field and its potential to benefit continuous health monitoring systems.

1.2 Research Aims

The primary research objectives of this study are as follows:

1. This project aims to showcase the ability of computational models to detect anomalies in movement data collected from radar without relying on the use of multiple sensors or wearables. This study also aims to demonstrate the viability of using radar technology in this domain.
2. Develop and evaluate the performance of various supervised and unsupervised learning techniques for anomaly detection. This project aims to develop computational models such as the Gaussian Mixture Model, Autoencoder, Convolutional Neural Network and more to identify the most suitable approach for this domain.
3. Compare the performance of unsupervised and supervised anomaly detection approaches on the collected radar dataset. By evaluating the strengths and limitations of each approach, this project aims to provide insights into the optimal strategies for detecting anomalies in human movement patterns.
4. Provide insights into radar technology's effectiveness in assisted living and digital healthcare for monitoring and detecting anomalies in human movement.

1.3 Thesis Structure

This thesis is split into five chapters as follows:

Chapter 1 introduces the motivation and problem statement, highlighting the need for an effective and non-intrusive method for detecting anomalies in human movement patterns. It also outlines the research aims and provides an overview of the thesis structure.

Chapter 2 presents a comprehensive background on human behaviour representation, data analysis techniques, and anomaly detection methods in the context of human activity monitoring systems. It reviews related work on machine learning and deep learning approaches for human activity recognition and anomaly detection.

Chapter 3 describes the methodology used in this project, including the experimental setup for data collection, data representation techniques used, and the construction of the dataset for analysis. It also demonstrates the methodology for developing and evaluating supervised and unsupervised computational models.

Chapter 4 presents the results of the experiments conducted. It outlines the various evaluation metrics and the corresponding discussions for the computational models and compares the performance of all the models tested.

Chapter 5 concludes the thesis by summarizing the project's essential findings and contributions. It highlights the effectiveness of radar technology and the proposed computational models in detecting anomalies in human movement patterns. This chapter also discusses the study's limitations and provides suggestions for future research.

Chapter 2

Background

2.1 Data Collection

In the field of movement quality analysis for assisted living, researchers employ diverse methods and protocols to collect data on human movement. Most existing scientific studies in this domain rely on wearable or non-wearable devices for data collection [12]. Data collected through non-wearable devices can further be classified into obtrusive visual sensing and non-obtrusive type dense sensing [4]. Obtrusive visual sensing techniques [40] utilize various kinds of cameras, such as infrared (IR), RGB [25], and depth cameras [11], to capture data. Despite the widespread use of visual sensing in this field, it presents several drawbacks, including varying illumination conditions, detecting shadows, and, most crucially, privacy concerns [5]. Similarly, the effectiveness of wearable devices is contingent upon the user consistently wearing the device, which may not always be practical. Another proposed solution involves using the sensors embedded in a user's smartphone [1]; however, this approach may only be feasible in some cases, as it requires the user to carry their smartphone at all times.

Non-obtrusive dense sensor networks [9] offer a promising alternative for data collection without compromising user privacy. This method of data collection uses various sensors, such as radio-frequency identification (RFID) [54], motion sensors, contact sensors, pressure sensors, and door sensors [24], which are deployed in the user's environment. For instance, the authors of [51] used motion detectors, pressure mats, break beam sensors and contact switches to perform activity recognition. Similarly, in [14], the authors used pressure sensors, motion sensors, door sensors and IoT-enabled technologies to combine individual historical data and environmental data to create regular activity profiles for patients with dementia. These profiles were used to detect anomalies and identify changes in a participant's health and well-being.

In the field of human activity recognition (HAR), some researchers have utilized pre-existing human activity datasets instead of collecting data directly from users. For example, the authors of [2] use two datasets from the CASAS project, Aruba and Cairo, to train their models. These datasets provide an excellent baseline regarding the daily activities of elderly patients in a smart home. They contain data collected from various sensors such as motion, temperature, door closure, etc. The data was collected for over

200 days and labelled with 11 and 13 activities, respectively.

2.1.1 Data Collection using Radar

Existing research clearly shows a need for effective, non-obtrusive contactless sensing methods. Radar technology provides a method that is not limited by the drawbacks of other methods, such as cameras [48]. It offers a sophisticated method for collecting data on movement patterns within a designated area. Through advanced algorithms, radar systems can transform raw electromagnetic reflections into detailed point clouds representing objects' spatial and temporal dynamics within the observed space. The radar then performs target localization and tracking on the point clouds. It uses a Kalman Filter variant to estimate the tracked targets' position, velocity, and acceleration in a 3D Cartesian space [45]. This technology has been growing in popularity in the field of human health monitoring. Many studies have used it in human activity recognition [20], [17], [10], fall detection [55], and gait analysis [43], [33].

2.2 Data Processing and Analysis

Appropriate data processing is crucial for classifying human activities or detecting anomalies. The data collected from various sensors is typically temporally ordered, making classifying activities directly or detecting anomalies challenging. Therefore, relevant features must be extracted from the raw data before any computational analysis.

According to [4], the data pre-processing and analysis process can be divided into two stages. The first stage, known as the lower sensory level, involves a series of pre-processing steps, including data filtering, segmentation, feature extraction and feature engineering. Data filtering helps reduce noise and remove any incorrect values in the data, ensuring the dataset's quality. Segmentation allows splitting the data into more manageable formats, such as smaller windows, facilitating using the sliding window approach [27] during classification.

Feature extraction is the process of extracting relevant features from the raw data and transforming them into numerical representations that computational models can process [34]. On the other hand, feature engineering involves using existing information from the raw data to create new features that can provide additional insights for the computational models during the learning process. Creating new features can help models better capture the underlying patterns and relationships within the data and improve classification or anomaly detection performance.

Feature normalization is an integral part of data pre-processing, mainly when dealing with features that have different scales or units. One popular technique for feature normalization is the Min-Max scaling, which linearly transforms the features to a specified range, typically between 0 and 1. The MinMaxScaler, a class provided by the scikit-learn library in Python, is widely used to apply Min-Max scaling to features [39]. This is crucial when using machine learning algorithms sensitive to feature scales, such as support vector machines or non-negative matrix factorization.

The collected data may exhibit an inherent bias in some cases due to the imbalanced distribution of data instances across the different class labels. For example, in anomaly detection tasks, it is expected to have more data instances representing normal behaviour than abnormal behaviour. This imbalance in data samples can lead to biased computational models [28], a phenomenon known as the “class imbalance problem.”

Several techniques can be used to mitigate the effect of class imbalance. One of the most popular approaches is resampling. This involves either oversampling the minority class or undersampling the majority class to balance the dataset. Oversampling techniques, such as the Synthetic Minority Over-sampling Technique (SMOTE), generate synthetic examples of the minority class to increase its representation in the dataset. On the other hand, undersampling techniques involve randomly removing instances from the majority class to reduce its dominance in the dataset. Studies have demonstrated that resampling techniques can improve the generalizability of learning algorithms in the presence of class imbalance [3].

The second stage of the analysis process is called the higher sensory level [4]. At this stage, the output from the lower level is further analyzed from a higher-level perspective. One essential task at the higher sensory level is anomaly detection, which involves identifying data instances that deviate from regular patterns. This is achieved by training models using normal data instances, allowing the models to establish a baseline for normal data. When a trained model encounters a new data instance that significantly deviates from the training data, it is classified as abnormal. These anomaly predictions can be further analyzed at a higher level to derive meaningful insights and take appropriate actions. For instance, when abnormal activity is detected in digital healthcare, such as falling or unusual movement gait, the system can notify emergency services or caregivers to ensure timely intervention and assistance.

2.2.1 Statistical Features for Movement Data

When dealing with temporally ordered movement data, it usually involves a time series of the subject’s position, acceleration and velocity values. This raw data can be used to extract a set of statistical features in the time and frequency domain [23], [32]. The most common statistical features in the time domain include the raw acceleration and velocity data’s mean, median and standard deviation values. Other features such as the minimum value, maximum value, the difference of the maximum and minimum values ($maximum - minimum$), interquartile range and the direction of movement can also be extracted. Similarly, features such as mean, standard deviation, median, maximum, minimum, $maximum - minimum$ value, and the energy can be extracted in the frequency domain.

2.2.1.1 Interquartile Range

The interquartile range (IQR) is a measure of statistical dispersion, representing the range between the first quartile (25th percentile) and third quartile (75th percentile) of a dataset. To extract the IQR feature from accelerometry data, the first quartile ($Q1$) and third quartiles ($Q3$) are calculated using the formulas given in equation 2.1, where n is the number of data points. Finally, the IQR can be computed as shown below in 2.2:

$$Q1 = \frac{(n+1)}{4}\text{th term}\ , \quad Q3 = \frac{3(n+1)}{4}\text{th term} \tag{2.1}$$

$$IQR = Q3 - Q1 \tag{2.2}$$

2.2.1.2 Direction Angle of Acceleration and Velocity

The direction angle of the acceleration and velocity can be determined using the acceleration and velocity values in the X and Y directions by calculating the arctangent of the ratio between the Y and X components. The formula to compute the direction angle (θ) is given in Equation 2.3, where a_y and a_x are the acceleration values, and v_y and v_x are the velocity values in the Y and X directions, respectively.

$$\theta = \arctan(\frac{a_y}{a_x}), \quad \theta = \arctan(\frac{v_y}{v_x}) \tag{2.3}$$

2.2.1.3 Energy in Frequency Domain

The energy in the frequency domain can be calculated after applying the Fast Fourier Transform (FFT) on the X and Y acceleration or velocity values. First, compute the FFT of the X and Y components separately, yielding the complex frequency-domain representations FFT_x and FFT_y. Then, calculate the energy (E) by summing the squared magnitudes of the complex FFT coefficients for both X and Y components as given in Equation 2.4, where N is the number of FFT points and k is the frequency bin index.

$$E = \sum_{k=0}^{N-1} |FFT_x[k]|^2 + |FFT_y[k]|^2 \tag{2.4}$$

2.2.2 Dimensionality Reduction

Dimensionality reduction is a crucial step in data analysis. It helps simplify high-dimensional data by transforming it into a lower-dimensional space while preserving the underlying structure and patterns of the high-dimensional data. The t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear dimensionality reduction technique that is particularly effective in visualizing high dimensional data in the 2D or 3D space. It aims to capture the local structure of the data by preserving the similarity between data points in the original high-dimensional space and the reduced low-dimensional space. This technique can be used to create visually interpretable representations of high-dimensional data that reveal hidden patterns, clusters and outliers. This technique has also been widely used in the field of human activity recognition. In [13], the authors use t-SNE to visualize human activities in the low-dimensional space. In [46], the authors proposed an ensemble learning framework for human activity recognition using smartwatches. They improved its accuracy to 96% using the t-SNE dimensionality reduction method.

2.3 Anomaly Detection

Anomaly detection in movement patterns or activities is challenging due to the rare occurrence of anomalies in the data [56]. According to [4], anomalies can be classified into three distinct types: point anomalies, contextual anomalies, and collective anomalies.

Point anomalies are the most straightforward type, representing a single data instance that deviates from the normal pattern. Anomaly detection for point anomalies involves setting a threshold value. If a new incoming data instance exceeds this threshold, it is classified as an anomaly. On the other hand, contextual anomalies are data instances that would not be considered abnormal on their own but are anomalous based on contextual information such as temporal and spatial features. For example, making tea at 3 AM would be considered an anomaly, even though making tea itself is normal. Collective anomalies are a set of data instances considered abnormal as a collection, even though each instance may not be anomalous. An example of a collective anomaly is a deviation from a regular sequence of activities.

Anomaly detection models aim to identify data instances that deviate from the normal patterns. The authors of [4] discuss two primary forms of anomaly detection: profiling and discrimination. Profiling involves training models using normal data instances and classifying any new data instances that deviate from this normal data as an anomaly. The approach followed in [49] falls under this category. In contrast, discriminating involves training models on abnormal behaviour and searching for similar patterns in new incoming data instances. The approaches used in [57] fall into this category. Due to the scarcity of anomalous data instances, profiling is a more realistic approach to anomaly detection and is more widely used than discriminating.

2.4 Computational Models

Various methodologies have been implemented for anomaly detection, including statistical, probabilistic, and machine-learning techniques. These methods aim to identify instances that deviate from the normal patterns in the data. Two main approaches are used when computationally analysing data: supervised learning and unsupervised learning.

Supervised learning is used when the collected data instances have known ground truth labels. This process is commonly referred to as activity recognition [59]. In this approach, the computational models are trained using labelled data, and the aim is to learn the mapping between the input features and the corresponding labels, enabling the model to classify new, unseen instances accurately.

On the other hand, unsupervised learning is used when the data instances do not have known labels. This process is commonly known as activity discovery [50]. In this method, computational models aim to identify underlying patterns and structures in the data without relying on defined labels. These methods help uncover hidden patterns, group similar instances, and identify anomalies based on their deviation from the discovered patterns.

2.4.1 Unsupervised Models

The notion that activities exhibit discernible ”regular patterns” has gained widespread acceptance among researchers in the field of anomaly detection [29]. These patterns can be effectively identified through unsupervised probabilistic models. In [6], the authors proposed a profiling strategy based on the Gaussian Mixture Model (GMM) to capture the normal behaviour of the occupants. The GMM is an unsupervised probabilistic model that estimates the likelihood of activities belonging to the normal data. The authors also argued that the GMM-based approach offers significant advantages over previously used Histogram models as the GMM could better capture the activity attribute's dependency.

While the GMM has proven effective in anomaly detection, other researchers have suggested that Hidden Markov Models (HMM) may be more suitable for noisy domains, such as smart homes. HMMs are probabilistic models that capture temporal dependencies and transitions between different states or activities. They have been successfully applied in various human activity recognition tasks [58], [52]. However, the use of HMMs also has limitations. For instance, the authors of [22] pointed out that HMMs may struggle to identify similar activity sequences accurately, mainly when the sequence duration differs significantly from the normal patterns.

Non-negative matrix factorization (NMF) is another unsupervised learning technique that has been applied to anomaly detection tasks. NMF is a dimensionality reduction and unsupervised learning technique that decomposes a non-negative matrix into two non-negative matrices. Non-negative matrix factorization (NMF) aims to find an approximation of a non-negative matrix $X \in R^{m\times n}$ as the product of two non-negative matrices $W \in R^{m\times k}$ and $H \in R^{k\times n}$, such that:

$$X \approx WH \tag{2.5}$$

The goal is to minimize the reconstruction error between X and WH, typically using a cost function such as the Frobenius norm:

$$||X - WH||_F \quad where, \quad W \geq 0, H \geq 0 \tag{2.6}$$

The matrix W contains basis vectors, which represent the latent features or patterns in the data, while the matrix H contains the corresponding coefficients or activations, indicating the contribution of each basis vector to the original data.

In the context of anomaly detection, NMF can be used to extract latent features from the data during the profiling phase. NMF captures the essential patterns and characteristics of normal behaviour by learning a low-dimensional representation of the normal data. In [15], the authors applied NMF to extract latent features for urinary tract infection (UTI) detection.

Accelerometer data has been widely used for human activity recognition (HAR) tasks, as it provides valuable information about individuals' movement and orientation. The authors of [31] introduced the Human Activity Recognition Trondheim (HARTH) dataset, which consists of accelerometer data collected from participants performing twelve different activities. They evaluated the performance of seven machine learning models for activity recognition on this dataset. Among the models tested, the authors found that unsupervised models, such as the support vector machine (SVM) and the random forest model, yielded promising results.

In another study, the authors of [53] proposed an ensemble detection model for anomaly detection in activities of daily living (ADL). The ensemble model consisted of four components: a one-class SVM (OC-SVM), an isolation forest (iForest), a local outlier factor (LOF), and a robust covariance estimator. The combination of these techniques achieved an accuracy of 98%.

2.4.2 Supervised Models

Supervised models have also been extensively used for human activity recognition and anomaly detection tasks. In [31], the authors evaluated the performance of several supervised models, such as Bidirectional Long Short-Term Memory (Bi-LSTM), Convolutional Neural Network (CNN), and Multi-Resolution CNN, on the HARTH dataset. These models have shown promising results in capturing the temporal dependencies and spatial features present in the accelerometer data.

In [2], the authors employed a Deep Neural Network (DNN) for activity classification, an overcomplete-deep autoencoder (OCD-AE) for anomaly detection within each activity class and an LSTM for predicting the next activity in a sequence. The authors then combined all three capabilities into a unified model to help patients with dementia.

Multimodal wearable activity recognition has also been explored using a combination of CNN and LSTM models. In [38], the authors introduced a deep learning architecture called DeepConvLSTM, which combines convolutional and recurrent layers to capture both spatial and temporal dependencies in the sensor data. This approach leverages the strengths of both CNN and LSTM models to improve activity recognition accuracy. The authors of [41] trained a CNN on the time series data collected from smartphone sensors to classify activities.

Chapter 3

Methodology

3.1 Overview

Radar technology is being explored in the field of anomaly detection for walking patterns due to its ability to preserve user privacy [36]. However, the effectiveness of this technology in this domain is still being investigated. Unlike previous studies that rely on multiple sensors or wearables to collect data, this project focuses on utilizing a single isolated radar in a fixed location within the environment. This study aims to demonstrate the functionality of radar-based anomaly detection in walking patterns and evaluate the efficiency of this technology. The project can be divided into the following parts:

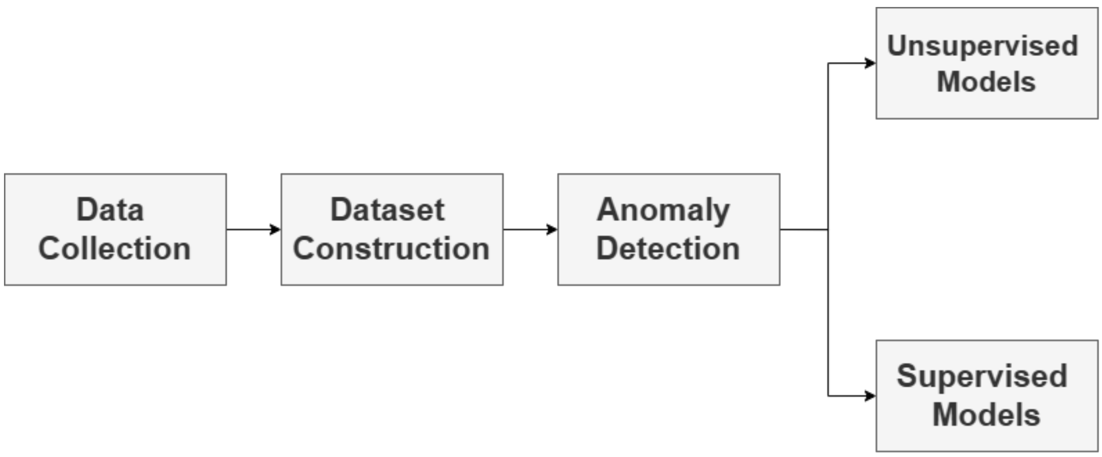

Figure 3.1: Project Structure

In the first part of the project, subjects were asked to perform a series of activities in a simulated environment. During these activities, a radar was set up in a fixed location to collect data. Both normal and simulated abnormal data were collected. All participants who volunteered for this experiment signed a consent form, which is included in appendix E. In the second part of the project, the collected data was pre-processed into a desired format for analysis. This step included data filtering, cleaning, and outlier removal to prepare the data for computational analysis. Finally, in the third part, various computational models were used to evaluate the efficiency of anomaly detection on the processed radar data. At this stage, two approaches were evaluated: unsupervised and supervised learning techniques. Table 3.1 lists the various computational models assessed in this study. Once the computational models were trained, their performance was evaluated using unseen data. All data processing, data analysis and computational model evaluation were done using Python on Jupyter Notebooks.

Unsupervised Models	Supervised Models
Non-Negative Matrix Factorization Model (NMF)	Convolutional Neural Network (CNN)
Gaussian Mixture Model (GMM)	Long Short-Term Memory (LSTM)
One-Class Support Vector Machine (OC-SVM)	CNN + LSTM Combined
Autoencoder (AE)	

Table 3.1: Computational Models Evaluated

3.2 Data Collection

The data collection process was conducted in the Laboratory for Robotic Assisted Living (LARA) at the National Robotarium at Herriot-Watt University. Seven subjects participated in the study, performing activities simulating an average day of their lives. The activities included making tea, making a sandwich, visiting the toilet, watching television, reading a magazine, playing with LEGO blocks, and using their mobile phones.

The AWR6843 mmWave radar from Texas Instruments was used during the data collection process. It was set up in the corner of the room on a tripod at a height of 160cm and rotated to the right at an angle of 30°. This ensured a good view of the entire room. The left side of Figure 3.2[1] shows the floor plan of the room[2], and the right side of Figure 3.2 shows a picture of the room from the viewpoint of the radar.

Each subject's data was collected three times in 55-minute intervals. Two sets of data were collected to show normal walking patterns; one set was collected to show simulated abnormal walking patterns. To demonstrate abnormalities in the walking pattern, subjects were instructed to simulate pain in their lower abdomen while moving around. Furthermore, subjects were asked to vary their walking speed during the abnormal dataset collection, especially while visiting the toilet.

To provide more context for researchers, the start and end times of each activity performed by the subjects were noted. Moreover, an RGB camera was set up next to the radar to capture the data collection process. This camera was built using a Raspberry Pi Model 3 and a Pi Camera. The web interface for the camera was built using an

[1]Floorplan edited from original illustration drawn by Saber Mirzaee Bafti

[2]Image is not drawn to scale and is not perfectly accurate. It only provides a better understanding of the structure of the room

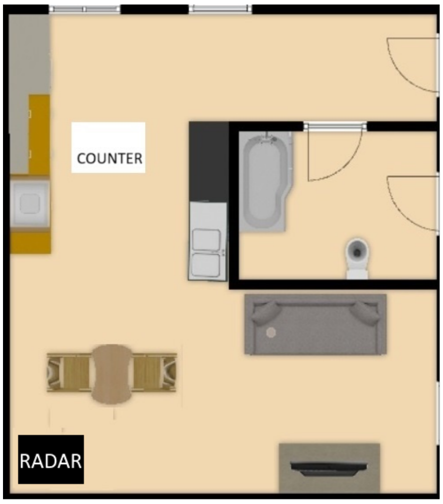

Figure 3.2: Left: Floor Plan of Room used for Data Collection. Right: The room from the viewpoint of the radar

open-source GitHub repository [35]. This allowed researchers to better understand the differences in the subjects' walking patterns.[3]

3.3 Data Representation

Unlike the previous data representation methods, which involve using sensors in the environment, this project relied on data collected from radar in a fixed location. The radar provided time series data that captured the subject's X, Y, and Z positions. Moreover, the radar could also extract the subject's X and Y accelerations and velocities.

The positional data collected can be used to plot heat maps that illustrate the areas where the subject spent the most time. These areas can be correlated to the various activities performed during the data collection process, providing valuable insights into the subject's behaviour. Figure 3.3 shows the heat map of the positional data of Subject D. From the figure, it is visible that the subject spent a considerable amount of time at the table and on the couch. This is associated with the activities of reading a magazine and watching television, respectively.

3.4 Dataset Construction

The raw data obtained from the radar included eight columns. Table 3.2 shows an example of a data sample. The first step in constructing the dataset is filtering the raw data and removing incorrect values. This process addressed issues where the radar could not localize the subject correctly, resulting in negative positional values. Similarly, data instances with positional values greater than 800 were removed, as they were outside the room's boundaries.

[3]The video footage from the camera was used solely by the researchers to aid in data processing and was deleted promptly after.

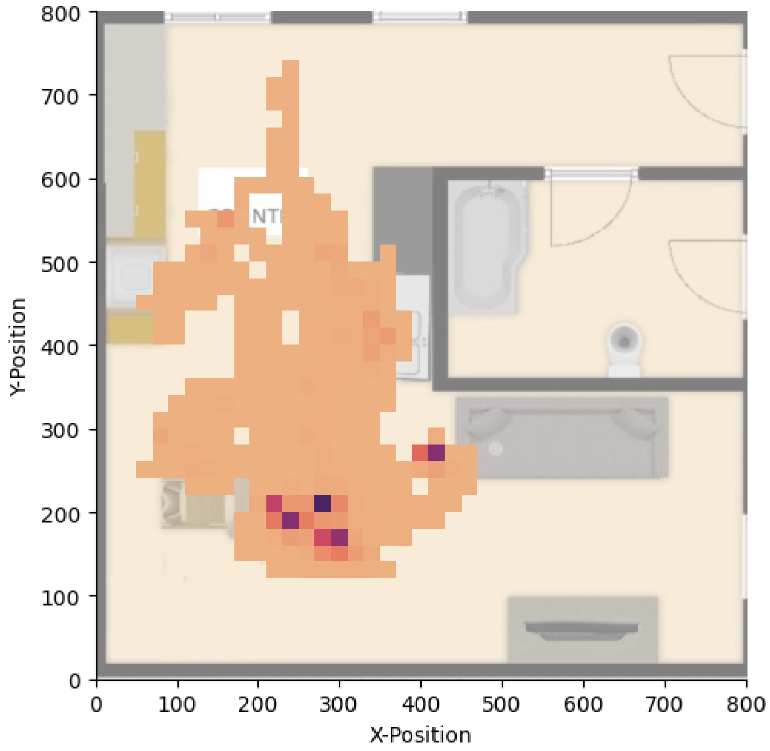

Figure 3.3: Heat Map of Normal Data of Subject D

X	Y	Z	AccX	AccY	VelX	VelY	Timestamps
268.39	133.68	60.05	-0.39	-0.26	-0.07	0.39	2023-12-14 09:40:39:607

Table 3.2: Example Data Sample

Since the main focus of this project is to detect anomalies in the walking patterns of a subject, data samples corresponding to the subject being stationary were excluded from the dataset. To achieve this, the difference in the Euclidean distance between each subsequent data sample was calculated. Data samples were removed if the Euclidean distance difference was less than 1, indicating no movement or very minute jerk movements. Furthermore, the X, Y and Z positional columns were removed from the data set. This is because the primary concern of this study is the movement quality rather than positional changes. Once the data was cleaned and filtered, feature extraction techniques were applied to extract relevant information from the dataset before computational analysis.

3.5 Unsupervised Models

Unsupervised techniques are always preferred over supervised approaches in real-world scenarios where the objective is to detect anomalies in movement patterns. This is because ground truth labels indicating whether a given data sample is normal or abnormal are unknown beforehand [7]. Therefore, it is crucial to develop methods to differentiate between normal and abnormal data without relying on pre-defined labels. This project employs a profiling strategy to perform anomaly detection using unsupervised models. This involves training computational models using normal data and identifying instances deviating from the normal patterns.

3.5.1 Feature Extraction

Before training the unsupervised models on normal data, relevant features were extracted from the cleaned time series data. This allows the models to understand better and capture underlying patterns in the data. This process involved slicing the data into smaller windows based on a chosen time interval. For example, if a time interval of 60 seconds were selected, each 55-minute dataset would be divided into 55 1-minute windows. For each sliced window, statistical features were extracted from the data. These statistical features capture various measures and characteristics of the acceleration and velocity magnitudes within each window [37]. Table 3.3 outlines the features extracted from the data in the time domain.

Moreover, features in the frequency domain were extracted as well. This process involved applying the Fast Fourier Transform to the X and Y acceleration and velocity values and extracting statistical features in the frequency domain. Table 3.4 outlines the features extracted from the data in the frequency domain. [4]

Acceleration Magnitude Features	Velocity Magnitude Features
Mean	Mean
Standard Deviation	Standard Deviation
Median	Median
Interquartile Range	Interquartile Range
Minimum	Minimum
Maximum	Maximum
Maximum-Minimum	Maximum-Minimum
Direction Angle	Direction Angle

Table 3.3: Time Domain Features Extracted for Unsupervised and Supervised Models

Acceleration Frequency Features	Velocity Frequency Features
Mean	Mean
Standard Deviation	Standard Deviation
Median	Median
Minimum	Minimum
Maximum	Maximum
Maximum-Minimum	Maximum-Minimum
Energy	Energy

Table 3.4: Frequency Domain Features Extracted for Unsupervised and Supervised Models

[4] These frequency domain features were only used to improve the performance of the OC-SVM Model and all the supervised models. All other unsupervised models only used the features in the time domain.

After extracting the features for each dataset, the feature values were normalized using the MinMaxScaler package from the scikit-learn library in Python [39]. This process ensures that all features contribute equally to the model's learning process and prevents features with larger magnitudes from dominating the others. A feature matrix can be generated for each dataset using the features from each window. Figure 3.4 shows the normal and abnormal data matrices for Subject D. These normalized features are processed further for each computational model into the desired format.

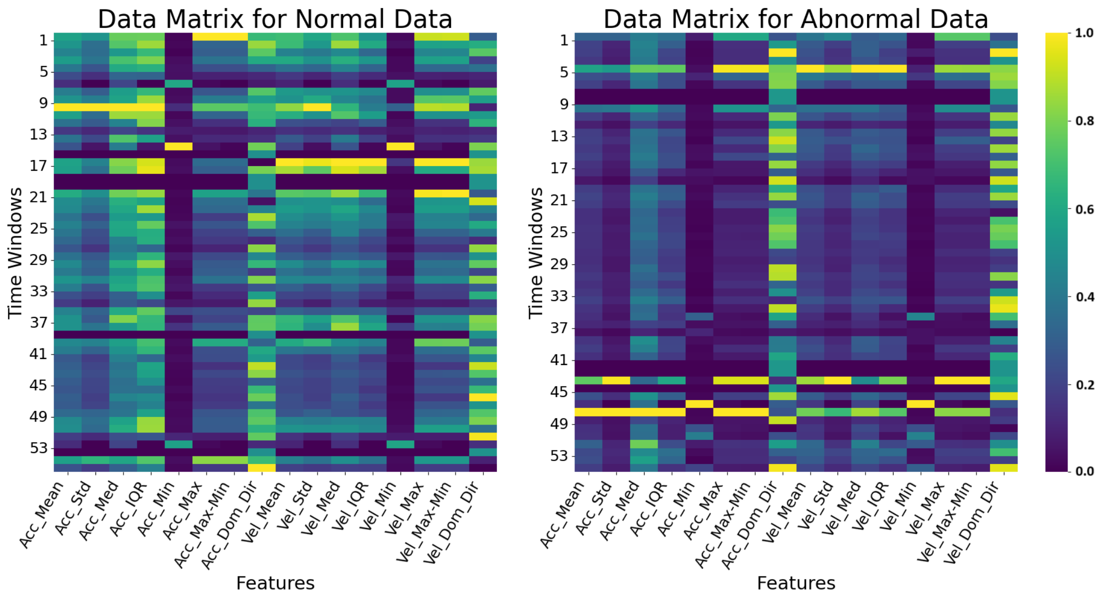

Figure 3.4: Normal and Abnormal Data Matrices for Subject D

3.5.2 Non-Negative Matrix Factorization (NMF) Model

The NMF algorithm is a powerful unsupervised learning technique for anomaly detection. It decomposes high-dimensional data into low-dimensional latent representations, capturing the inherent patterns and structures in the data. By learning the system's normal behaviour, NMF can effectively identify deviations and anomalies, making it a robust choice for anomaly detection in complex datasets.

3.5.2.1 Data Preparation

This study collected data from seven subjects, resulting in 14 normal and 7 abnormal datasets. After the data pre-processing and feature extraction process, 14 normal and 7 abnormal data feature matrices were obtained.

The 14 normal data matrices were split into three subsets: 8 training matrices, 3 validation matrices, and 3 test matrices to prepare the data for training and testing. The 8 training matrices were combined to form the training data for the model, and the 3 test matrices were combined with the 7 abnormal data matrices to form the test data for the model. This ensured that the model was tested on completely unseen data.

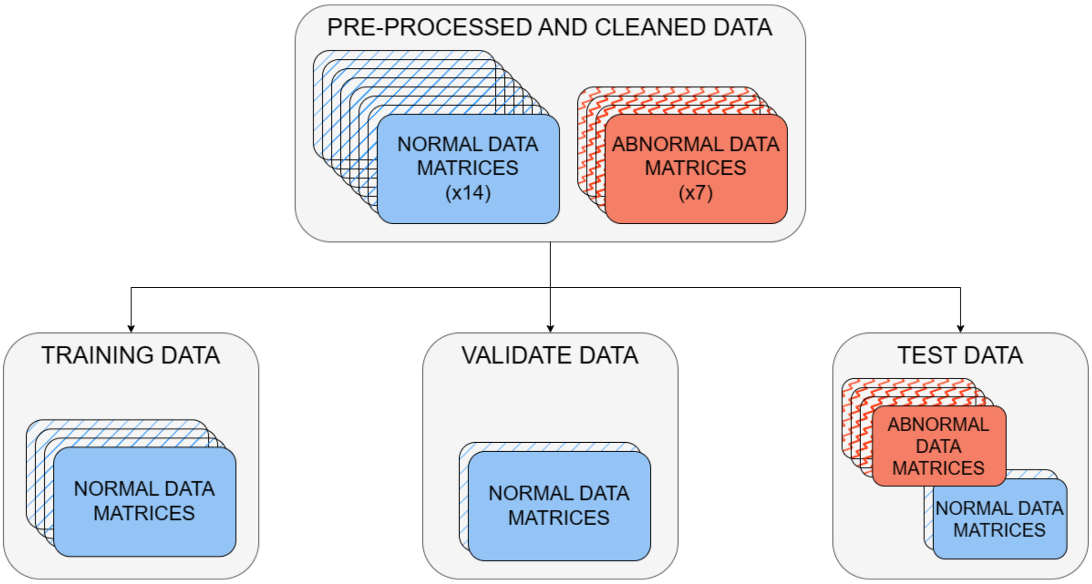

Figure 3.5: Division of Dataset for NMF Model

Figure 3.5 demonstrates the steps involved in splitting the pre-processed and cleaned data.

3.5.2.2 Anomaly Detection

The NMF model aims to decompose a data matrix into two smaller non-negative matrices. These decomposed matrices can be used to reconstruct the original data matrix, and the reconstruction error between the original and the reconstructed matrices can be measured. When the NMF model is trained on normal data, it learns the underlying patterns of the normal data. Once trained, the model should be able to reconstruct normal data matrices with low reconstruction error. Consequently, abnormal data matrices should be reconstructed with a higher reconstruction error.

To perform anomaly detection with an NMF model, the model is first fitted with the normal training data. Then, a threshold for the reconstruction error must be determined to classify a given data matrix as normal or abnormal. This project used the reconstruction error of the three validation data matrices to set the threshold value. Any data matrix with a higher reconstruction error than this threshold was considered an anomaly, while any matrix with a reconstruction error lower than this threshold was classified as normal.

3.5.3 Gaussian Mixture Model (GMM)

The GMM is a probabilistic approach commonly used for anomaly detection tasks. By modelling the normal data as a mixture of Gaussian distributions, the model can capture the underlying structure and density of the data. If a data instance deviates significantly from the learned distributions, it is classified as an anomaly.

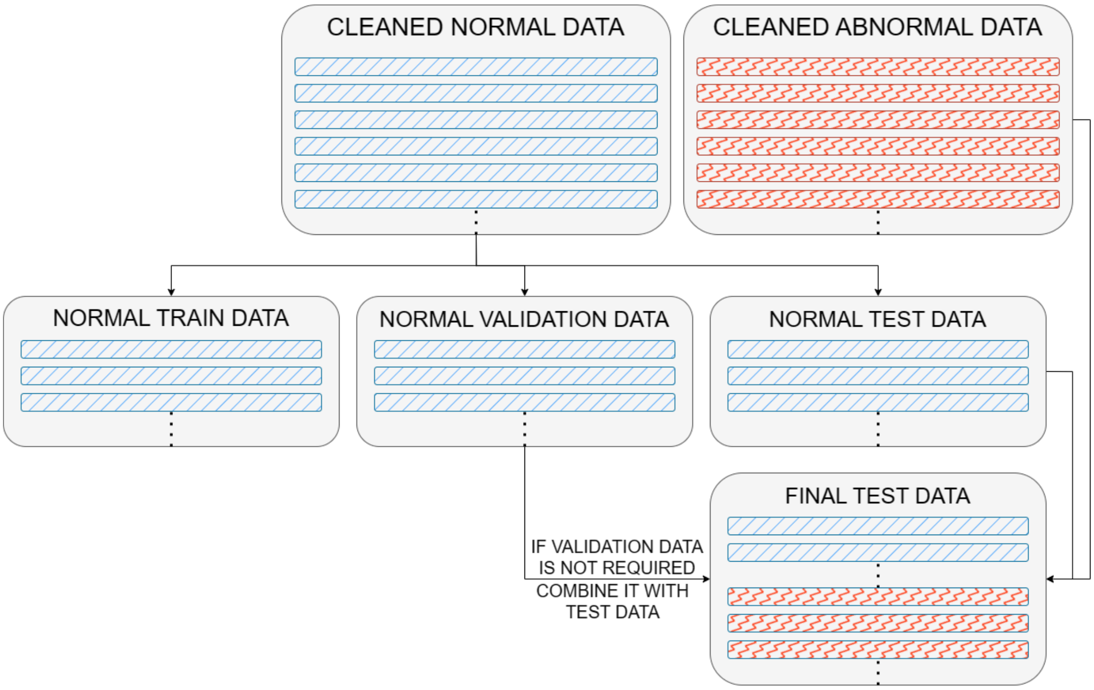

Figure 3.6: Division of Dataset for Unsupervised Models

3.5.3.1 Data Preparation

After the data pre-processing and feature extraction steps, there are 14 normal and seven abnormal datasets. All the normal datasets are combined to form a singular normal dataset, and similarly, all the abnormal data is concatenated to form a singular abnormal dataset. The normal dataset is then split into three subsets: train data, validation data, and test data, following an 80:10:10 split. The training data is used to train the GMM, and the test data is added to the abnormal data to get the final testing data.

Figure 3.6 shows the division of the dataset for this model.

3.5.3.2 Anomaly Detection

The GMM is trained using the normal training dataset to learn the underlying data distribution in the normal instances. The trained GMM consists of Gaussian components, each characterized by its mean, covariance, and mixing coefficient [18]. These components collectively represent the learned normal behaviour. The log-likelihood values of the data instances in the validation dataset are calculated to determine the threshold for anomaly detection. The maximum log-likelihood value from the validation set is chosen as the threshold value. Any new data instance with a log-likelihood below this threshold is classified as an anomaly, while instances with log-likelihood values above the threshold are classified as normal. By setting the threshold using validation data, overfitting the model can be avoided while ensuring that the model generalizes well to unseen data.

3.5.4 One-Class Support Vector Machine (OC-SVM) Model

The OC-SVM is an anomaly detection technique that learns a tight decision boundary around normal instances in high-dimensional feature spaces. It is particularly effective when most training data belongs to the normal class, and anomalies are rare or absent during training. By learning a hyperplane that maximally separates the normal instances from the origin, OC-SVMs create a decision boundary that encloses the normal data points.

3.5.4.1 Data Preparation

To train the OC-SVM model, all the normal datasets from the subjects are concatenated to form a single normal dataset after data pre-processing and feature extraction steps. Similarly, combining all the abnormal datasets creates a single abnormal dataset. The features extracted in the frequency domain were also used for this model. The normal dataset is then split into training and testing data using an 80:20 split. The testing data is appended to the abnormal dataset to produce the final testing data. This model did not require a validation dataset.

Figure 3.6 shows the division of the dataset for this model.

3.5.4.2 Anomaly Detection

The OC-SVM model learns a decision boundary that encloses most of the normal instances in the training data. The objective is to find a hyperplane that maximally separates the normal instances from the origin in the feature space. The OC-SVM model optimizes the decision boundary during training based on the normal training data. In the detection phase, the trained OC-SVM model classifies new instances based on their position relative to the learned decision boundary. Instances falling outside the boundary are considered anomalies, while those inside the boundary are classified as normal.

3.5.5 Autoencoder (AE) Model

An AE model is an unsupervised learning model that has been widely used for anomaly detection tasks [19]. The AE's ability to learn a compressed representation of normal data and reconstruct it with minimal error makes it well-suited for identifying anomalies. The model captures the inherent patterns and structures in the data by training an AE to minimize the reconstruction error in normal instances. Anomalies that deviate from the normal patterns tend to have higher reconstruction errors when passed through the trained AE.

3.5.5.1 Data Preparation

To train the AE model, all the normal datasets from the subjects are concatenated to form a single normal dataset. Similarly, combining all the abnormal datasets creates a single abnormal dataset. The normal dataset is then split into training, testing, and validation data using an 80:10:10 split. The testing data is appended to the abnormal dataset to produce the final testing data.

Figure 3.6 shows the division of the dataset for this model.

3.5.5.2 Anomaly Detection

The AE model consists of an encoder network that compresses the input data into a lower-dimensional latent space and a decoder network that reconstructs the original data from the latent representation. The specific architecture of the AE used in this project is as follows:

- Encoder:
  - 3 Dense layers with decreasing number of neurons
  - Bottleneck Layer
  - Each dense layer is followed by batch normalization and dropout layers
- Decoder:
  - 3 Dense layers with an increasing number of neurons
  - Each dense layer is followed by batch normalization and dropout layers

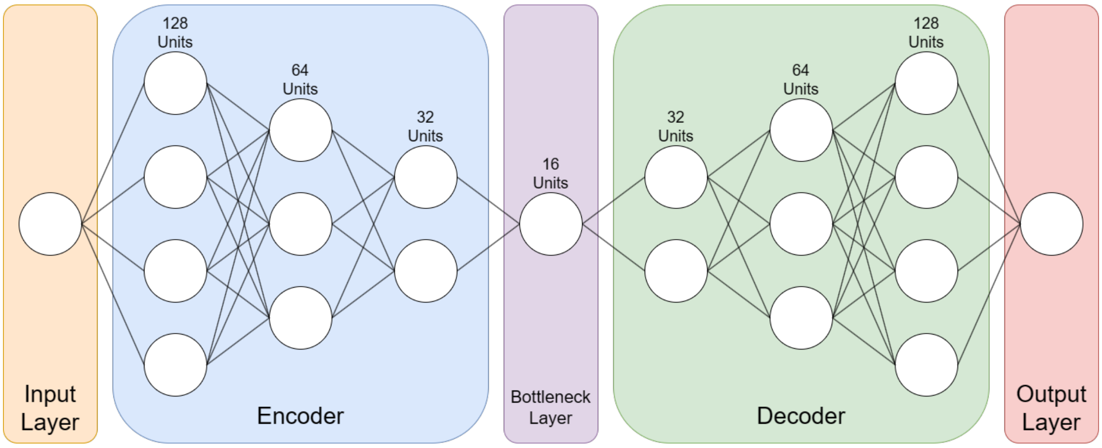

Figure 3.7: Autoencoder Model Architecture

Figure 3.7 provides a visual representation of the model architecture used for the autoencoder.[5]

During the training phase, the AE learns to minimize the reconstruction error between the input and the reconstructed output. The training involves feeding the normal data instances through the encoder to obtain the compressed latent representation and then reconstructing the original data using the decoder. After training, a threshold reconstruction error must be determined for anomaly detection. The validation set is

[5] Although not explicitly shown in the image, each dense layer is followed by batch normalization and dropout layers.

used to set this threshold. The reconstruction errors of the validation instances are calculated, and a suitable threshold is selected.

The trained AE model is evaluated using the test dataset during the testing phase. Each test instance is passed through the AE, and its reconstruction error is computed. The instance is classified as anomalous if the reconstruction error exceeds the predetermined threshold. Conversely, the instance is classified as normal if the reconstruction error is below the threshold.

3.6 Supervised Models

Supervised models, such as convolutional neural networks (CNN) and long short-term memory (LSTM) models, were evaluated in this project to compare their efficiency and performance with unsupervised approaches. Although supervised models require labelled data, which may be scarce or costly to obtain, their inclusion in this study provides a benchmark for evaluating the effectiveness of unsupervised techniques. It highlights the potential benefits of incorporating labelled data when available.

3.6.1 Feature Extraction

In the field of anomaly detection, the effectiveness of supervised models heavily relies on the quality of data and the features fed into the model while training [7]. To ensure that the data is suitable and ready for training and evaluation of supervised models, the cleaned time series normal data of all subjects are combined into a single dataset. Similarly, one dataset is constructed for the abnormal data. Ground truth labels are added to each data instance in both datasets, and additional relevant features are extracted, including[6]:

- Change in X and Y positions (dy and dx) between consecutive samples
- Euclidean distance covered between consecutive samples

At this stage, the normal and abnormal datasets are divided into two parts, train and test, each with an 80:20 split. The training data from the normal and abnormal datasets are combined to form the combined training data. Similarly, the combined testing data is formed. To capture the data's temporal nature, a sliding window approach is used on this data [27].[7] The time series data is divided into windows of specific size. The window is moved sequentially across the data using a pre-determined stride value. A set of statistical features is extracted for each window, both in the time and frequency domains. Table 3.3 outlines the features extracted in the time domain, while Table 3.4 shows the features extracted in the frequency domain. The extracted features were normalized for each window using the MinMaxScaler from the scikit-learn Python package [39].

[6] These features provide the model with information on the relative change in position, indicating the speed of movement and not position directly

[7] The data was split into train and test sets before window extraction to avoid data leakage while testing as the sliding window approach can potentially lead to data being used for training and testing.

Figure 3.8: An Example of the Sliding Window Approach

Figure 3.8 shows the extraction of windows from the dataset. This sliding window approach was used for both the train and test sets.

Anomaly detection tasks often face the challenge of class imbalance, where normal data instances significantly outnumber abnormal data instances. To address this issue, the data is resampled to balance the number of data instances in both classes. To perform data resampling, two methods were used; the first method involves synthetically oversampling the minority class using the synthetic Minority Over-sampling Technique (SMOTE). This creates artificial anomalies to balance the training and test data. The second method involves randomly undersampling the majority class by removing random data instances from the majority class to balance the dataset. For data resampling, the smote, and RandomUnderSampler functions were used from the imbalanced-learn Python package [30]. The performance of both methods was compared.

Finally, the balanced train and test sets are used to train and evaluate the model's performance respectively. This method ensures that there is no overlap between the train and test sets and that the model is evaluated on completely unseen data.

3.6.2 Convolutional Neural Network (CNN) Model

Using a Convolutional Neural Network (CNN) model for this task is motivated by its proven effectiveness in learning spatial and temporal patterns from time series data [16]. CNNs have demonstrated remarkable performance in various domains, including anomaly detection, due to their ability to extract hierarchical features and capture local dependencies automatically.

The architecture of the CNN model used in this study is designed to learn features from the input time series data and classify them as normal or abnormal. The model consists of three convolutional layers, each followed by a dropout layer and a max-pooling layer. The convolutional layers apply learned filters to the input data, capturing local patterns and extracting relevant features.

Dropout layers are incorporated after each convolutional layer to prevent overfitting and improve the model's generalization capability [44]. Dropout layers regularise the model by randomly dropping out a fraction of the neurons during training and reducing reliance on specific features. Max-pooling layers help the model focus on the most prominent features and reduce computational complexity.

After the convolutional and pooling layers, the extracted features are flattened and passed through two fully connected dense layers. These layers learn high-level representations and capture complex relationships among the features [19]. Dropout is also applied between the dense layers to regularize the model further.

Finally, the output layer consists of a single neuron with a sigmoid activation function, producing a probability score between 0 and 1. This score represents the likelihood of an input sample being anomalous. Figure 3.9 visually represents the model architecture.

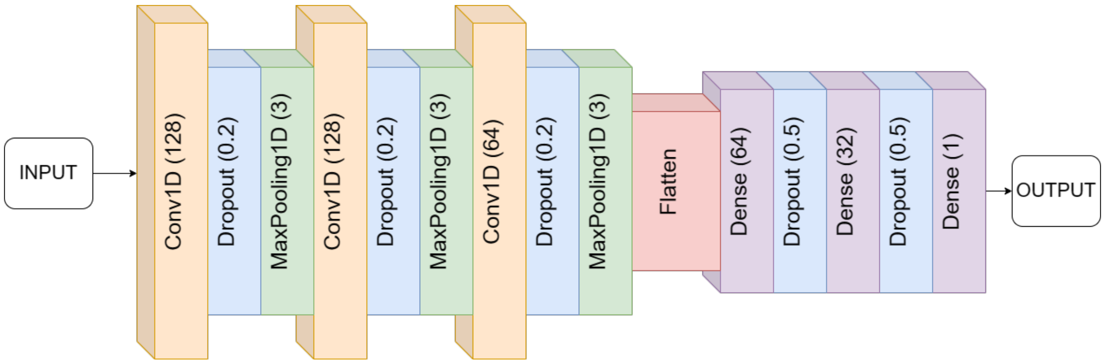

Figure 3.9: Model Architecture of Convolutional Neural Network

3.6.3 Long Short-Term Memory (LSTM) Model

LSTM models are widely used to capture long-term dependencies and temporal patterns in sequential data [21]. LSTMs are a type of recurrent neural network (RNN) that have shown exceptional performance in modelling time series data and have been successfully applied to anomaly detection tasks [47].

The model's architecture used in this study consists of two LSTM layers, two dense layers and an output layer. The first LSTM layer has 64 units and returns sequences, allowing the model to capture the temporal dynamics of the input data. The second LSTM layer has 32 units and returns only the last output, providing a compact representation of the entire sequence. Dropout layers are added after each LSTM layer to regularize the model and prevent overfitting. The output from the second LSTM layer is then passed through two dense layers to help the model learn higher-level representations and capture complex relationships among the learned features. Dropout is also applied after each dense layer to regularize the model further.

Finally, the output layer consists of a single neuron with a sigmoid activation function, producing a probability score between 0 and 1. This score indicates the likelihood of an input sample being anomalous. Figure 3.10 visually represents the model architecture used.

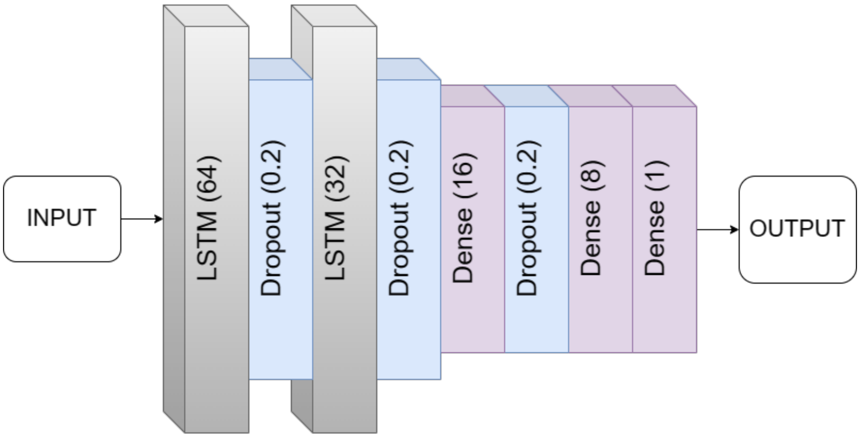

Figure 3.10: Model Architecture of the Long Short-Term Memory

3.6.4 CNN + LSTM Combined Model

The motivation behind exploring a combined CNN-LSTM model is to leverage the strengths of both architectures in capturing spatial and temporal patterns from time series data [26]. CNNs have proven effective in extracting local features and learning hierarchical representations, while LSTMs excel at modelling long-term dependencies and temporal dynamics [38].

The model consists of three convolutional layers, followed by two bidirectional LSTM layers and two dense layers. The convolutional layers apply learned filters to the input data, capturing local patterns and extracting relevant features. Dropout layers are incorporated after each convolutional layer to prevent overfitting and improve the model's generalization capability. Max-pooling layers are employed to downsample the feature maps, reducing the spatial dimensions and focusing on the most prominent features.

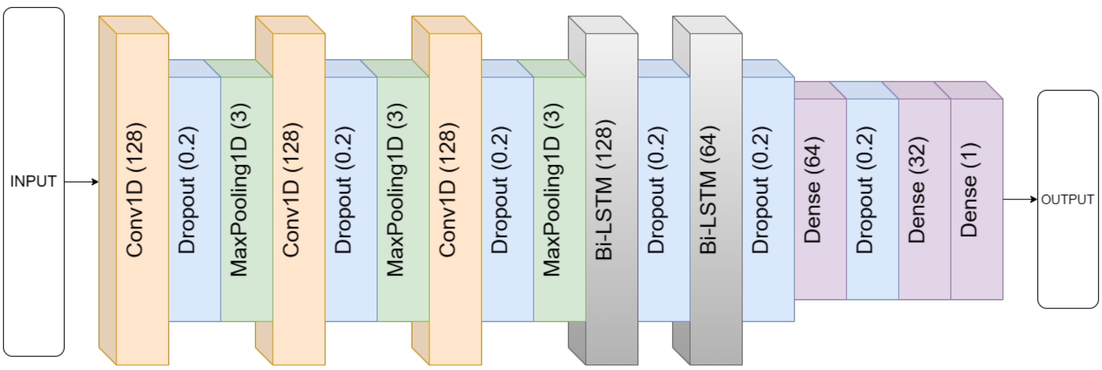

Figure 3.11: Model Architecture of CNN+LSTM Combined

The output from the convolutional layers is then passed through two bidirectional LSTM layers. Bidirectional LSTMs process the sequence in both forward and backward directions, allowing the model to capture dependencies from past and future contexts [42]. The first bidirectional LSTM layer has 128 units and returns sequences, while the second bidirectional LSTM layer has 64 units and returns only the last output. Dropout layers are added after each bidirectional LSTM layer to regularize the model and prevent overfitting.

The output from the second bidirectional LSTM layer is then passed through two fully connected dense layers. Finally, the output layer consists of a single neuron with a sigmoid activation function, producing a probability score between 0 and 1. This score indicates the likelihood of an input sample being anomalous. Figure 3.11 visually represents the model architecture used.

Chapter 4

Results

4.1 Evaluation Method

In this project, the performance of the tested anomaly detection models is evaluated using various metrics based on the prediction outcomes. These outcomes can be categorized into four types:

- True Positives (TP): The model correctly identifies anomalous instances as anomalies.
- True Negatives (TN): The model correctly identifies normal instances as normal.
- False Positives (FP): The model incorrectly identifies normal instances as anomalies.
- False Negatives (FN): The model incorrectly identifies anomalous instances as normal.

Using these values, a general set of evaluation metrics can be calculated. They are:

- Accuracy: This is measured as the ratio of correctly classified outcomes to the total number of predicted outcomes. It measures the overall correctness of the model's predictions.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.1}$$

- Precision: This metric evaluates the accuracy of the model's positive predictions. It is calculated as the ratio of true positives (TP) by the total number of instances predicted as positive by the model. It indicates how well a model avoids false positives.

$$Precision = \frac{TP}{TP+FP} \tag{4.2}$$

- Recall: Recall measures the model's ability to correctly identify positive instances from the test data. It is calculated as the ratio of true positives (TP) to the total number of actual positive instances.

$$Recall = \frac{TP}{TP+FN} \tag{4.3}$$

- F1-Score: This metric is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. It accounts for both false positives and false negatives, which is very useful in tasks where the classes are imbalanced, such as anomaly detection.

$$F1-Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{4.4}$$

4.2 Unsupervised Models

Unsupervised learning techniques play a crucial role in anomaly detection, as they can identify patterns in normal data instances without the need for ground-truth labels. This section presents the results of the unsupervised models mentioned in Table 3.1. Each model was evaluated based on its ability to capture the underlying structure of normal data and effectively classify abnormal data instances. The results demonstrate the strengths and limitations of each model, providing valuable insights into their use in anomaly detection tasks.

4.2.1 Non-Negative Matrix Factorization Model

The first step in evaluating the NMF model was to identify a time sample value that would allow us to split the normal datasets into smaller segments. After rigorous testing, the time sample of 80 seconds was chosen for this model. This time sample demonstrated the best performance in capturing the underlying patterns of the normal data and effectively identifying abnormal data instances. Figure A.1 shows the three cleaned datasets of Subject D with an 80-second time sample. From this image, it is visually evident that there is a difference between the normal and abnormal data.

Before testing the trained model, a threshold reconstruction error was chosen. The validation dataset was used to find the optimal threshold value, and the trained model was used to find the validation dataset's reconstruction error. This error value was chosen as the threshold for anomaly detection. Data matrices with a reconstruction error higher than this threshold were classified as anomalies, while those with a lower error were classified as normal data instances.

The NMF model was evaluated using sevral metrics, including accuracy, precision, recall and F1-score. Table 4.1 presents the evaluation results for the 80 second time sample. The model achieved an accuracy of 90%, indicating its overall effectiveness

Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
90	91.25	90	89.33

Table 4.1: Evaluation Metrics for NMF Model

in classifying normal and abnormal data. The precision score of 91% suggests that the model has a low false positive rate, while the recall score of 90% indicates that the model successfully identifies a high proportion of actual anomalies. The F1 score of 89.3% provides a balanced measure of the model's performance, considering both precision and recall. Figure 4.1 shows the reconstructions errors of the test matrices in comparison to the chosen threshold value.

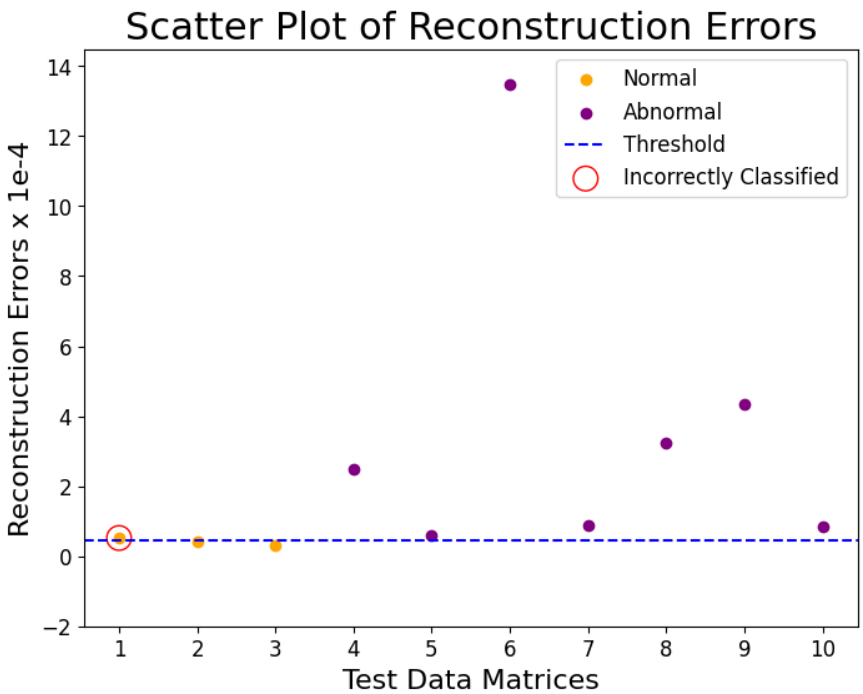

Figure 4.1: Reconstruction Errors of the NMF Model

Since the test set of this model has a very small subset of data, K-Fold cross-validation was also used to evaluate this model. The model was trained and evaluated using 10 folds. Table 4.2 shows the evaluation metrics after implementing the K-Fold cross-validation.

Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
86.53	78.87	86.53	82.44

Table 4.2: Evaluation Metrics for NMF Model with 10-Fold Cross Validation

4.2.1.1 Discussion

The results demonstrate the effectiveness of the NMF model in detecting anomalies in movement patterns. The selected time sample and threshold value proved optimal for distinguishing between normal and abnormal data instances. The high accuracy and f1-score demonstrate that the model can make accurate predictions while maintaining a good balance between precision and recall, making it a reliable model for anomaly detection in movement patterns. Moreover, the results obtained from the 10-fold validation method further validate the model's performance. The data used to train this model involved multiple subjects with varying styles of movement and speeds. Yet, the model could learn the underlying patterns and generalize to unseen subjects. The cross validation method further strengthens the performance of the model and its ability to generalize well to unseen data, which is crucial for its practical application.

However, this approach also has limitations. This approach for anomaly detection performs well for data extracted over a considerable interval of time. For instance, in this project, 55-minute datasets were used to train the NMF model. Although the dataset was split into smaller segments, the model was trained using all of the smaller segments of each dataset. This provided the model with more information to learn underlying patterns from. This model might not perform as well if smaller data sets are used. This implies that performing real-time anomaly detection for a subject would be difficult. However, if data is collected over a longer time, this approach is an effective anomaly detection method.

4.2.2 Gaussian Mixture Model Model

To train the Gaussian Mixture model, a similar approach to that used to the NMF model was used. First, a time sample had to be determined. After performing an in-depth analysis with a wide range of time samples, 130 seconds was chosen. This time sample rate yielded the best performance in capturing the underlying structure of the normal data and effectively classifying abnormal data instances. Once the model is trained, it outputs the log-likelihood values of a new data instance being a part of the normal distribution.

To perform anomaly detection, a threshold log-likelihood had to be determined. The validation dataset was used to choose the optimal value. From all the log-likelihood values of the validation dataset, the maximum log-likelihood value was chosen as the threshold. New data instances with a log-likelihood less than this threshold were classified as anomalies, while those with a higher log-likelihood value were classified as normal data instances.

The GMM model was evaluated using the accuracy, precision, recall and F1-score metrics. Table 4.3 outlines the evaluation metrics for the GMM. The GMM achieved an

Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
85.52	87.66	85.52	80.3

Table 4.3: Evaluation Metrics for GMM Model

accuracy of $\approx 85\%$ indicating that it can differentiate normal data from abnormal data fairly well. The precision score of $\approx 87\%$ shows that the model has a relatively low false positive rate, while the recall score of $\approx 85\%$ suggests that the model successfully identifies a substantial proportion of actual anomalies.

4.2.2.1 Discussion

The results demonstrate the effectiveness of using a Gaussian Mixture Model in detecting anomalies in movement patterns. The time sample and threshold chosen proved to yield good performance by the model. From the accuracy and f1-score, it is clear that the model can effectively perform anomaly detection while maintaining a balance between precision and recall. Like the NMF model, the GMM is also effective in generalizing data instances of different subjects. This is a very desirable quality of a well-performing model in this domain.

Moreover, unlike the NMF model, this model can identify anomalies in a shorter duration. Since the chosen time sample for this model is 130 seconds, this trained GMM should be able to identify anomalies in data samples of 130 seconds. This provides a better method of anomaly detection that can be tuned further to use in real time.

4.2.3 One-Class Support Vector Machine Model

An optimal time sample had to be selected to split the normal data into smaller, more manageable segments to train the OC-SVM model. Rigorous testing was conducted to determine the optimal time sample rate for the OC-SVM model. Various time sample rates were evaluated, and their impact on the model's performance was assessed. The time sample that best-balanced model performance and computational complexity were chosen. After testing, this was identified as 30 seconds.

Furthermore, different kernel functions were explored to identify the most suitable one for the OC-SVM model. The tested kernels included linear, polynomial, radial bases function (RBF) and sigmoid. The performance of each kernel was evaluated and compared using the evaluation metrics. The RBF kernel demonstrated the best accuracy and was selected as the optimal kernel for this domain. However, the other kernels also have their merits.

The OC-SVM model with different kernels was evaluated using the accuracy, precision, recall and f1-score metrics. Table 4.4 presents the evaluation metrics for the OC-SVM model with the different kernels.

Kernel Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Linear	68.24	60.19	68.24	62.09
Polynomial	69.35	62.31	69.35	63.42
RBF	69.53	60.09	69.53	61.78
Sigmoid	67.96	58.51	67.96	61.06

Table 4.4: Evaluation Metrics for OC-SVM Model

4.2.3.1 Discussion

The results demonstrate that the OC-SVM exhibits varying performance in detecting anomalies, depending on the choice of the kernel function. Among the kernel functions, the RBF kernel displays high accuracy and recall values of $\approx 70\%$. This indicates that the model effectively captures the underlying patterns of normal behaviour and identifies actual anomalies. However, the $\approx 60\%$ precision value suggests a relatively high false positive rate. This may lead to some normal instances being misclassified as anomalies.

The polynomial kernel also demonstrated comparable performance to the RBF kernel, with an accuracy and recall of $\approx 69\%$. However, it has a higher precision score of $\approx 62\%$, indicating a lower false positive rate when compared to the RBF kernel. The linear and sigmoid kernels exhibit similar performance on almost all metrics, but the linear kernels performs slightly better on all metrics.

The choice of the kernel function plays a crucial role in the performance of the OC-SVM model in anomaly detection. The RBF and polynomial kernels perform better than the linear and sigmoid kernels. This suggests that the RBF and polynomial kernels are more suitable for capturing the complex and non-linear patterns present in the movement data. However, despite the RBF kernel achieving the highest accuracy and recall, its lower precision score indicates a trade-off between correctly identifying anomalies and minimizing false positives.

It is important to note that the performance of the OC-SVM model still needs to improve compared to some of the other unsupervised anomaly detection models evaluated in this study. The inherent complexity and variability in movement patterns across subjects prove challenging for the OC-SVM model. Nevertheless, the OC-SVM model, particularly with the RBF or polynomial kernel, can serve as a complementary approach to another anomaly detection technique. Further research and exploration could improve the performance of the OC-SVM model in detecting anomalies in movement patterns collected with a radar.

4.2.4 Autoencoder Model

A wide range of samples were evaluated to determine the optimal time sample for the AE model. The impact on the model's performance was assessed every time the sample was tested. After thorough experimentation, the time sample of 40 seconds was chosen as it yielded the best performance.

Furthermore, several hyperparameters were tuned to optimize the performance of the AE model. The activation function for all the dense layers was evaluated, and the 'tanh' function was identified to yield the best results. A grid search also tuned the learning rate and number of epochs. A model with a learning rate 1e-05 combined with the Adam optimizer was trained for 1000 epochs to yield the best performance. Moreover, L2 regularization with a value of 0.01 was applied to each dense layer to mitigate overfitting. Furthermore, checks for early stopping and overfitting detection were set up to prevent training beyond the required capacity.

The AE model was evaluated using the accuracy, precision, recall and f1-score metrics. Table 4.5 provides an outline of the evaluation metrics for the AE model for all the activation functions tested. The best AE achieved an accuracy of $\approx 56\%$, indicating moderate effectiveness in classifying normal and anomalous instances. Notably, the model demonstrated a high precision of $\approx 85\%$, suggesting a low false positive rate.

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ELU	52.91	83.22	52.91	61.86
ReLU	51.84	85.1	51.84	60.68
Tanh	55.52	84.88	55.52	64.04

Table 4.5: Evaluation Metrics for AE Model

This means that when the model classifies a data instance as an anomaly, it is highly likely to be a true anomalous instance. However, the moderate recall score of $\approx 56\%$ indicates that the model might miss some actual anomalies, resulting in a higher false negative rate.

4.2.4.1 Discussion

The results demonstrate that the model can effectively identify anomalies, especially when trying to minimise the false positive rate, which can be crucial depending on the application. Moreover, it can be seen that the model with the ReLu activation yields a slightly better precision score than the model with the Tanh activation. However, the other metrics are better for the model with the Tanh activation.

Although the model demonstrates a high precision score, the model only shows moderate accuracy and recall scores. This can be attributed to several factors. Firstly, the inherent complexity and variability in the movement patterns across subjects could pose a challenge to the model. The AE model may struggle to capture all the nuances and variations present in the data, leading to some anomalies being missed.

Secondly, the limited depth of the AE architecture may restrict the model's capacity to learn intricate patterns. While deeper models were explored, they consistently led to overfitting, indicating that the available data might not be sufficient to support a more complicated architecture. To improve the model's performance, future research can involve more diverse training data or incorporate ensemble models to combine the strengths of multiple models.

4.3 Supervised Models

Supervised learning is a powerful technique for anomaly detection when labelled data is available. These models learn from examples of normal and abnormal data instances to effectively identify new instances as normal or abnormal. In this section, the results of the evaluated supervised models are presented. Each model was trained on labelled data using a sliding window approach. These results show the strengths of supervised models in the domain of anomaly detection in movement patterns of data collected through a radar.

All of the models evaluated followed a sliding window approach. This involved splitting the data into train and test sets first. Then, the data samples were split into smaller windows based on a chosen window length. The windows were extracted by sliding the window across the data with a chosen stride value. This allowed the training and testing data to effectively capture the temporal nature of the data. After thorough experimentation, a window size of 200 and a stride value of 50 were chosen. A window size of 200 is approximately equal to 20 seconds of data collected. These values were selected to balance the available data while capturing enough information in each window to perform anomaly detection accurately.

Moreover, all the models were trained for 200 epochs and optimized using the Adam optimizer. Checks for early stopping and overfitting detection were set up to ensure a model was not trained beyond the required capacity. At the end of training, the model with the best weights was chosen for evaluation.

It is important to note that data collected with the radar was imbalanced and contained more normal than abnormal instances. To better evaluate the supervised models, the collected data was resampled to balance the dataset. Two approaches are evaluated: synthetic oversampling of the minority class and random undersampling of the majority class. The results of both methods are presented.

Multiple activation functions were evaluated in all the supervised models tested. The functions evaluated were the ELU, ReLU, and Tanh activation functions for the dense and convolutional layers. For a particular model evaluated, all the layers in that model used the same activation function unless specified otherwise.

4.3.1 Convolutional Neural Network

The evaluation metrics of the CNN model are presented below. Table 4.6 presents the evaluation metrics for all the activation functions when the minority class was synthetically oversampled using the SMOTE function from the ‘imbalanced-learn’ Python package [30]. Table 4.7 presents the evaluation metrics for all activation functions when the majority class was randomly undersampled using the RandomUnderSampler function from the ’imbalanced-learn’ Python package [30].

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ELU	88.38	86.38	91.16	88.7
ReLU	83.56	88.33	77.35	82.47
Tanh	86.05	81	94.2	87.1

Table 4.6: Evaluation Metrics for the CNN Model with Minority Class Oversampling

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ELU	82.7	75.74	96.22	84.76
ReLU	82.43	81.25	84.32	82.76
Tanh	85.41	81.04	92.43	86.37

Table 4.7: Evaluation Metrics for the CNN Model with Majority Class Undersampling

4.3.2 Long Short-Term Memory

The evaluation metrics of the LSTM model are presented below. Table 4.8 presents the evaluation metrics for both the activation functions when the minority class was synthetically oversampled using the SMOTE function. Table 4.9 presents the evaluation metrics for both activation functions when the majority class was randomly undersampled using the RandomUnderSampler function.

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ELU	81.77	90.21	71.27	79.63
ReLU	81.77	90.21	71.27	79.63
Tanh	81.22	96.31	64.92	77.56

Table 4.8: Evaluation Metrics for the LSTM Model with Minority Class Oversampling

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ELU	78.92	72.2	94.05	81.69
ReLU	88.1	88.95	87.03	87.98
Tanh	86.49	86.1	87.03	86.56

Table 4.9: Evaluation Metrics for the LSTM Model with Majority Class Undersampling

4.3.3 CNN + LSTM Combined Model

The evaluation metrics of the CNN+LSTM model are presented below. Table 4.10 presents the evaluation metrics for the activation functions when the minority class was synthetically oversampled using the SMOTE function. Table 4.11 presents the evaluation metrics for the activation functions when the majority class was randomly undersampled using the RandomUnderSampler function.[1]

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ELU	81.63	88.29	72.93	79.88
ReLU	83.98	90.73	75.69	82.53
Tanh	81.21	88.18	72.1	79.33
CNN-tanh, Dense-relu	88.95	88.95	88.95	88.95

Table 4.10: Evaluation Metrics for CNN+LSTM Model with Minority Class Oversampling

4.3.4 Discussions

The supervised models tested, namely, CNN, LSTM, and a hybrid CNN+LSTM combined model, demonstrate impressive performance in detecting anomalies in movement

[1] These tables only show the metrics of models with the same activation function for all layers and the best-performing model that used more than one activation function. The evaluation metrics of all the models can be seen in Table B.1 and Table B.2 of the appendix

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ELU	83.51	77.43	94.59	85.16
ReLU	85.4	84.66	86.49	85.56
Tanh	85.13	78.51	96.76	86.68
CNN-tanh, Dense-elu	88.1	84.06	94.05	88.76

Table 4.11: Evaluation Metrics for CNN+LSTM Model with Majority Class Undersampling

patterns. Tables 4.6 to Table 4.11 showcase the effectiveness of these models with different activation functions and data resampling methods.

When the minority class is synthetically oversampled using SMOTE, the CNN+LSTM model achieves the highest accuracy, precision, recall and F1-score with the convolutional layers using a tanh activation and the dense layers using a relu activation. Most of the models with other activation functions have significantly lower metrics. The LSTM model achieves an accuracy of $\approx$81% and an F1-Score of $\approx$79% across all the activation functions. However, the tanh activation has a slightly lower performance than the other two activation functions. The similar performance across the activation functions for the LSTM model can be attributed to the activation function only being applied to the dense layers in the model. The CNN model with the elu activation function performs almost as well as the CNN+LSTM model, while the CNN model with the other two activations has a lower performance.

When the majority class was randomly undersampled, the models maintained high performance. Once again, the CNN+LSTM model achieved the highest accuracy and F1-Score amongst all models. The best-performing model used a tanh activation for the convolutional layers and an elu activation for the dense layers. The LSTM model showed a drastic difference in performance when the majority class was randomly undersampled. The performance of the models with the relu and tanh activation functions increased when compared to when the data was synthetically oversampled. This can be attributed to the fact that synthetically oversampling the data might not contain the same temporal nature as the original data, but randomly undersampling the majority class can preserve the original temporal nature of the data. However, the model with the elu activation had significantly lower performance when the data was undersampled compared to when the data was oversampled. The CNN model showcased similar performance across both resampling techniques; however, when the data was undersampled, the best-performing model used the tanh activation. The performance of the CNN model across the other two activation functions was similar.

The impressive performance of the supervised models can be attributed to many factors. Firstly, the feature extraction process, including features from the time and frequency domains, provides a comprehensive representation of the data, capturing relevant patterns. Including additional features such as the Euclidean distance, dy, and dx further enhances the model's ability to learn patterns in the data.

Secondly, the high performance can also be attributed to the resampling techniques. The data resampling techniques ensure the dataset is balanced, allowing for an equal distribution of normal and abnormal instances. This ensures that the model is trained equally in both cases and prevents the model from favouring one class over the other. Furthermore, using SMOTE to oversample the minority class yielded the best overall performance. This can be attributed to the fact that using this technique for data resampling generates more training data for the model to learn from. Randomly undersampling the normal data reduces the overall size of the dataset, resulting in less data for the model to learn from.

The t-SNE dimensionality reduction technique was used to further support these evaluation metrics. The data can be plotted on a scatter plot by reducing the high-dimensional data to two dimensions. Figure 4.2 shows the high-dimensional data for the training and test sets plotted in two dimensions.

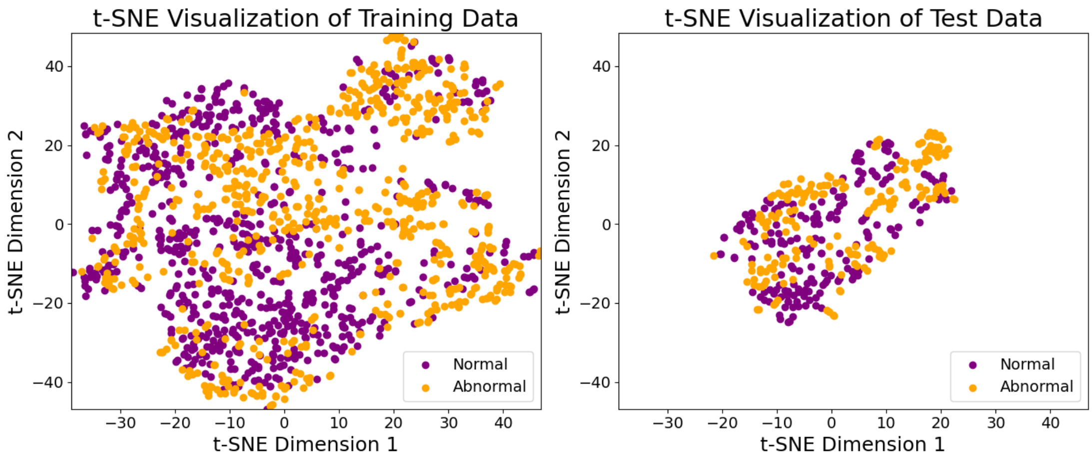

Figure 4.2: t-SNE Dimensionality Reduction of Training Data

Based on the scatter plot, it can be observed that there is some level of distinction between the two classes in the reduced dimensional feature space, although not very clear. This indicates that the model is able to detect some inherent patterns within the data but not with complete accuracy. Moreover, there are multiple areas where the classes overlap, which suggests that there are some challenges in making precise classifications. This observation is also reflected in the evaluation metrics of the models. This difficulty in accurate classification is expected as the data collected is highly likely to be similar across both class labels due to the simulated nature of the abnormal walking patterns.

Chapter 5

Conclusions

5.1 Summary

This study demonstrates radar technology's viability in detecting anomalies in movement patterns. It develops and evaluates anomaly detection models for movement patterns using data collected from a single radar in a fixed location. This allows for anomaly detection in movement patterns without infringing on user privacy or needing multiple sensors or wearables.

The data collection process involved seven subjects performing activities in a simulated environment at the Laboratory for Robotic Assisted Living (LARA). Each subject's data was collected three times, in 55-minute intervals, with two sets representing normal walking patterns and one set representing abnormal walking patterns. The AWR6843 mmWave radar from Texas Instruments was used to collect the data.

The collected data was pre-processed by filtering, cleaning and removing outliers in the data to prepare it for analysis. This study evaluated both unsupervised and supervised approaches to identify anomalies in movement patterns from the radar data effectively.

Before running computational analysis with unsupervised models, feature extraction was performed. The cleaned radar data was sliced into smaller windows based on a chosen time interval. For each window, a set of statistical features was extracted from the acceleration and velocity values in the time domain. Statistical features were also extracted in the frequency domain for the One-Class Support Vector Machine Model. The extracted features were then normalized using the MinMaxScaler.

Of the unsupervised models, the NMF model achieved the best performance with an accuracy of 90% and F1-Score of 89.3% when a time interval of 80 seconds was used. The GMM model, with a 130-second time sample, achieved an accuracy of 85% and F1-Score of 80.3%. The performance of the OC-SVM varied depending on the kernel function used, with the RBF kernel achieving the highest accuracy and recall of $\approx$70%. The AE model demonstrated a high precision of $\approx$85% but moderate accuracy and recall values of around 55%.

A different process was used to perform computational analysis using supervised approaches. A sliding window approach was used on the cleaned radar data to extract windows of length 200, using a stride of 50. This is approximately 20 seconds of data. For each window, feature extraction was performed. In the time domain, 14 statistical features were extracted from the acceleration and velocity values. In the frequency domain, 14 statistical features were extracted from the acceleration and velocity values. Features such as Euclidean distance and X and Y position change were also extracted. All the features were normalized using the MinMaxScaler.

Since the collected data contained more instances of normal data than abnormal data, it was resampled to balance the dataset. Resampling techniques such as SMOTE and RandomUnderSampler from the 'imbalanced-learn' Python package were tested [30]. The performance of all the models using both resampling techniques was evaluated.

When the minority class was oversampled, the CNN model with elu activation achieved an accuracy of $\approx$88% and an F1-Score of $\approx$89%. The LSTM model with elu and relu activations performed equally well with an accuracy of $\approx$82% and an F1-score of $\approx$80%. The CNN+LSTM model had the best performance with an accuracy and F1-score of $\approx$89%.

When the majority class was undersampled, the CNN model with the tanh activation yielded the best performance with an accuracy and F1-score of $\approx$85% and $\approx$86%, respectively. The LSTM model yielded an accuracy and F1-score of $\approx$88%. The CNN+LSTM model had an accuracy and F1-score of $\approx$88% and $\approx$89%, respectively.

This study highlights the effectiveness of both unsupervised and supervised models in detecting anomalies in movement patterns using radar data. The results show that supervised techniques perform better at anomaly detection than unsupervised approaches overall. This can be attributed to the availability of labelled data in this study. Moreover, the feature extraction process, data resampling techniques, and the combination of time and frequency domain features could have contributed to the performance of the supervised models.

In conclusion, this paper demonstrates the potential of using radar to detect anomalies in movement patterns. This method is a nonintrusive, privacy-preserving alternative to other widely used data collection methods. The study's findings support the use of radar technology in assisted living and digital healthcare domains to monitor and detect anomalies in human movement. The results also show that with enough research into the development of robust computational models, radar technology is a viable non-invasive privacy-preserving data collection alternative to existing methods.

5.2 Future Work

This study effectively demonstrates the viability of radar technology for anomaly detection in movement patterns. However, there are a lot of limitations to this study. Firstly, the data collected involves simulating pain in the lower abdomen while moving around. The data collection process should be repeated with subjects with known abnormal walking patterns in future works. This would provide a more accurate representation of anomalies in movement patterns. Moreover, there is a vast disparity in the number of ordinary to abnormal data instances. This imbalance in the dataset should be addressed in future works to allow computational models to learn underlying patterns in the normal data more effectively. This study included seven subjects in the data collection process. Increasing the number of participants will allow the model to have more data to learn patterns better while also better generalizing to the general population.

One possible scope of future work is predicting subjects given a data instance. This will further support the potential of radar technology in this domain. If computational models are able to accurately identify subjects from a data instance, they would be able to identify intricate patterns in the movement quality. This would also indicate that the model can perform anomaly detection with better performance.

Another scope for future work is the detection of contextual and collective anomalies. This study focused on point anomaly detection. However, contextual and collective anomaly detection is a field that can provide immense benefits to the assisted living and digital healthcare domain. Future studies can explore computational analysis involving contextual features such as position. For collective anomalies, future studies can focus on detecting anomalies in a subject's activity sequences.

5.2.1 Subject Prediction

This study also considered an initial exploration of subject prediction. This demonstrates the possibility of this work in the future and its potential to benefit this domain.

A supervised approach was followed to perform subject prediction. A CNN+LSTM model was used to identify a subject based on input from a normal data instance. To perform subject prediction, all the subjects' normal data were combined. The data was first split into test and train sets. Then, a sliding window approach was used to extract smaller windows of length 200 with a stride of 50. For each window, features in the time and frequency domain were extracted. Additional features used in the supervised anomaly detection were also extracted. For windows that had data instances of multiple subjects, the mode of the subjects in that window was chosen as the ground truth label for that window. The CNN+LSTM model was trained for 30 epochs using the Adam optimizer. Early stopping and overfitting detection measures were also used. After training, the model with the best weights was used for testing. Table 5.1 shows the

Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
55	30.25	55	39.03

Table 5.1: Evaluation Metrics for CNN+LSTM Model for Subject Prediction

evaluation metrics of this model for subject prediction. This model only demonstrates the ability of a computational model to perform subject prediction. The moderate performance of this model can be attributed to the lack of sufficient training data to accurately learn the differences between the movement patterns of each subject. In the future, this work can be improved further to yield better-performing models for subject prediction.

5.2.2 Activity Sequence Anomaly Detection

Detecting anomalies in activity sequences is extremely important in digital healthcare. This initial exploration of collective anomaly detection demonstrates an initial approach that can be used to evaluate the performance of collective anomaly detection models.

First, the normal sequence of activities must be extracted from the collected normal data to detect anomalies in activity sequences. In this initial approach, using the X and Y positions of the subject, four zones were defined in the room where the data was collected. Figure C.1 in the appendix visually represents this. The time series data was converted to a sequence of activities based on the zone where the subject was present at every data sample. An activity was only considered if the subject spent at least 2 seconds in a given zone consecutively. This ensured that randomly walking around multiple zones was not considered an activity in each zone. Moreover, multiple consecutive data instances in a single zone were regarded as only one activity instead of the same repeating activity over time.

Once the normal activity sequences for each subject were extracted, these activities were mapped to numerical values to ensure compatibility with computational models. Furthermore, a sliding window approach was used with this sequence of activities. A window of 6 activities was chosen, with a stride length of 4 activities. The normal data was then split into train, validation and test sets. The test normal data was combined with the abnormal data to produce the final testing data. The normal data was then used to train a Hidden Markov Model (HMM). When the trained HMM encounters a new data instance, it outputs a log probability score of the data instance being a part of the learned model. If the log probability is high, the data instance is classified as normal. If the log probability is low, then the data instance is classified as an abnormal data instance.

A threshold log probability score must be determined for anomaly detection using the HMM. The validation set was used to choose an optimal threshold. The truncated value of the log probability of the validation data was selected as the threshold. Using this threshold, the model was evaluated on the test data. Table 5.2 presents the evaluation metrics of this initial model. These metrics can be used as a proof concept to show that

Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
67.5	25	22.22	23.53

Table 5.2: Evaluation Metrics for HMM for Collective Anomaly Detection

radar data can be used for collective anomaly detection. However, the model's moderate performance indicates that more research needs to be conducted to accurately identify collective anomalies from radar data. In the future, various models can be explored to achieve better performance for collective anomaly detection.

Bibliography

[1] Nadeem Ahmed, Jahir Ibna Rafiq, and Md Rashedul Islam. Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model. *Sensors*, 20(1), 2020.

[2] Khaled A. Alaghbari, Mohamad Hanif Md. Saad, Aini Hussain, and Muhammad Raisul Alam. Activities recognition, anomaly detection and next activity prediction based on neural networks in smart homes. *IEEE Access*, 10:28219–28232, 2022.

[3] Fayez Alharbi, Lahcen Ouarbya, and Jamie A Ward. Comparing sampling strategies for tackling imbalanced data in human activity recognition. *Sensors*, 22(4), 2022.

[4] U. A. B. U. A. Bakar, Hemant Ghayvat, S. F. Hasanm, and S. C. Mukhopadhyay. *Activity and Anomaly Detection in Smart Home: A Survey*, pages 191–220. Springer International Publishing, Cham, 2016.

[5] Allah Bux, Plamen Angelov, and Zulfiqar Habib. Vision based human activity recognition: A review. In Plamen Angelov, Alexander Gegov, Chrisina Jayne, and Qiang Shen, editors, *Advances in Computational Intelligence Systems*, pages 341–371, Cham, 2017. Springer International Publishing.

[6] Fabien Cardinaux, Simon Brownsell, Mark Hawley, and David Bradley. Modelling of behavioural patterns for abnormality detection in the context of lifestyle reassurance. In José Ruiz-Shulcloper and Walter G. Kropatsch, editors, *Progress in Pattern Recognition, Image Analysis and Applications*, pages 243–251, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009.

[8] Chen Chen, Caifeng Shan, Ronald M. Aarts, and X. Long. Sensing and computing for smart healthcare. *J. Ambient Intell. Smart Environ.*, 14:3–4, 2021.

[9] Liming Chen, Jesse Hoey, Chris D. Nugent, Diane J. Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012.

[10] Qingchao Chen, Yang Liu, Bo Tan, Karl Woodbridge, and Kevin Chetty. Respira-

tion and activity detection based on passive radio sensing in home environments. *IEEE Access*, 8:12426–12437, 2020.

[11] Claudio Coppola, Serhan Cosar, Diego R. Faria, and Nicola Bellotto. Social activity recognition on continuous RGB-D video sequences. *International Journal of Social Robotics*, 12(1):201–215, 2020. Place: Germany Publisher: Springer.

[12] Samundra Deep, Xi Zheng, Chandan Karmakar, Dongjin Yu, Leonard G. C. Hamey, and Jiong Jin. A survey on anomalous behavior detection for elderly care using dense-sensing networks. *IEEE Communications Surveys Tutorials*, 22(1):352–370, 2020.

[13] Ramesh Dharavath, G. MadhukarRao, Himanshu Khurana, and D. Edla. t-sne manifold learning based visualization: A human activity recognition approach. pages 33–43, 2020.

[14] Shirin Enshaeifar, Ahmed Zoha, Andreas Markides, Severin Skillman, Sahr Thomas Acton, Tarek Elsaleh, Masoud Hassanpour, Alireza Ahrabian, Mark Kenny, Stuart Klein, Helen Rostill, Ramin Nilforooshan, and Payam Barnaghi. Health management and pattern analysis of daily living activities of people with dementia using in-home sensors and machine learning techniques. *PLOS ONE*, 13(5):1–20, 05 2018.

[15] Shirin Enshaeifar, Ahmed Zoha, Severin Skillman, Andreas Markides, Sahr Thomas Acton, Tarek Elsaleh, Mark Kenny, Helen Rostill, Ramin Nilforooshan, and Payam Barnaghi. Machine learning methods for detecting urinary tract infection and analysing daily living activities in people with dementia. *PLOS ONE*, 14(1):1–22, 01 2019.

[16] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

[17] Francesco Fioranelli and Julien Le Kernec. Radar sensing for human healthcare: challenges and results. In *2021 IEEE Sensors*, pages 1–4, 2021.

[18] Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[20] Sevgi Zubeyde Gurbuz and Moeness G. Amin. Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring. *IEEE Signal Processing Magazine*, 36(4):16–28, 2019.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[22] Ya-Xuan Hung, Chih-Yen Chiang, Steen J. Hsu, and Chia-Tai Chan. Abnormality detection for improving elder's daily life independent. In Yeunsook Lee,

Z. Zenn Bien, Mounir Mokhtari, Jeong Tai Kim, Mignon Park, Jongbae Kim, Heyoung Lee, and Ismail Khalil, editors, *Aging Friendly Technology for Health and Independence*, pages 186–194, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[23] Andrey D. Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Appl. Soft Comput.*, 62:915–922, 2018.

[24] Vikramaditya Jakkula and Diane Cook. Detecting anomalous sensor events in smart home data for enhancing the living experience. 01 2011.

[25] Ahmad Jalal, Yeon-Ho Kim, Yong-Joong Kim, Shaharyar Kamal, and Daijin Kim. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recogn.*, 61(C):295–308, jan 2017.

[26] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE Access*, 6:1662–1669, 2018.

[27] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 289–296, 2001.

[28] Jiawen Kong, W. Kowalczyk, S. Menzel, and Thomas Bäck. Improving imbalanced classification by anomaly detection. pages 512–523, 2020.

[29] Yu Chen Lee. Human activity recognition/monitoring and anomaly detection using radar sensor. M.sc. honours thesis, University of Edinburgh, Edinburgh, UK, 2023.

[30] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[31] Aleksej Logacjov, Kerstin Bach, Atle Kongsvold, Hilde Bremseth Bårdstu, and Paul Jarle Mork. Harth: A human activity recognition dataset for machine learning. *Sensors*, 21(23), 2021.

[32] Inês Machado, A. L. Gomes, H. Gamboa, V. Paixão, and Rui M. Costa. Human activity data discovery from triaxial accelerometer sensor: Non-supervised learning sensitivity to feature extraction parametrization. *Inf. Process. Manag.*, 51:204–214, 2015.

[33] Sophia Mancini, Willis Troy, Kerstyn Hall, Xinye Wu, and Henry Wang. Radar technology as a mechanism for clinical gait analysis: A review. *Journal of Annals of Bioengineering*, 2021, 12 2020.

[34] MathWorks. What Is Feature Extraction? https://www.mathworks.com/discovery/feature-extraction.html, 2024. [Accessed: 28-March-2024].

[35] Silvan Melchior. Rpi_cam_web_interface: A web interface for the rpi cam. https://github.com/silvanmelchior/RPi_Cam_Web_Interface, 2024.

[36] Antonio Nocera, Linda Senigagliesi, Gianluca Ciattaglia, and Ennio Gambi. Walking pattern identification of fmcw radar data based on a combined cnn and bi-lstm approach. In *2023 IEEE 36th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 275–280, 2023.

[37] Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-garadi, and Uzoma Rita Alo. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, 105:233–261, 2018.

[38] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1), 2016.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[40] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[41] Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, 59:235–244, 2016.

[42] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[43] Ann-Kathrin Seifert, Moeness G. Amin, and Abdelhak M. Zoubir. Toward unobtrusive in-home gait analysis based on radar micro-doppler signatures. *IEEE Transactions on Biomedical Engineering*, 66(9):2629–2640, 2019.

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[45] Texas Instruments. *Group Tracker Parameter Tuning Guide for the 3D People Counting Demo*, 2021. Rev 1.1.

[46] Dipanwita Thakur, A. Guzzo, and G. Fortino. t-sne and pca in ensemble learning based human activity recognition with smartwatch*. *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)*, pages 1–6, 2021.

[47] Kim Phuc Tran, Huu Du Nguyen, and Sébastien Thomassey. Anomaly detection using long short term memory networks and its applications in supply chain management. *IFAC-PapersOnLine*, 52(13):2408–2412, 2019. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019.

[48] Ingrid Ullmann, Ronny G. Guendel, Nicolas Christian Kruse, Francesco Fioranelli, and Alexander Yarovoy. A survey on radar-based continuous human activity recognition. *IEEE Journal of Microwaves*, 3(3):938–950, 2023.

[49] Gilles Virone, Majd Alwan, Siddharth Dalal, Steven W. Kell, Beverely Turner, John A. Stankovic, and Robin Felder. Behavioral patterns of older adults in assisted living. *IEEE Transactions on Information Technology in Biomedicine*, 12(3):387–398, 2008.

[50] Jiahui Wen and Mingyang Zhong. Activity discovering and modelling with labelled and unlabelled data in smart environments. *Expert Syst. Appl.*, 42:5800–5810, 2015.

[51] D. H. Wilson and C. Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In Hans W. Gellersen, Roy Want, and Albrecht Schmidt, editors, *Pervasive Computing*, pages 62–79, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[52] Tingting Xue and Hui Liu. Hidden markov model and its application in human activity recognition and fall detection: A review. In Qilian Liang, Wei Wang, Xin Liu, Zhenyu Na, and Baoju Zhang, editors, *Communications, Signal Processing, and Systems*, pages 863–869, Singapore, 2022. Springer Singapore.

[53] S.W. Yahaya, A. Lotfi, and M. Mahmud. Detecting anomaly and its sources in activities of daily living. *SN Computer Science*, 2(14), 2021.

[54] Lina Yao, Quan Z. Sheng, Xue Li, Tao Gu, Mingkui Tan, Xianzhi Wang, Sen Wang, and Wenjie Ruan. Compressive representation for device-free activity recognition with passive rfid signal strength. *IEEE Transactions on Mobile Computing*, 17(2):293–306, 2018.

[55] Yicheng Yao, Changyu Liu, Hao Zhang, Baiju Yan, Pu Jian, Peng Wang, Lidong Du, Xianxiang Chen, Baoshi Han, and Zhen Fang. Fall detection system using millimeter-wave radar based on neural network and information fusion. *IEEE Internet of Things Journal*, 9(21):21038–21050, 2022.

[56] Jie Yin, Qiang Yang, and Jeffrey Junfeng Pan. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1082–1090, 2008.

[57] Dong Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 611–618 vol. 1, 2005.

[58] Licheng Zhang, Xihong Wu, and Dingsheng Luo. Human activity recognition with hmm-dnn model. In *2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pages 192–197, 2015.

[59] Qingchang Zhu, Zhenghua Chen, and Y. Soh. A novel semisupervised deep learning method for human activity recognition. *IEEE Transactions on Industrial Informatics*, 15:3821–3830, 2019.

Appendix A

Data Matrices for 80 Second Time Samples

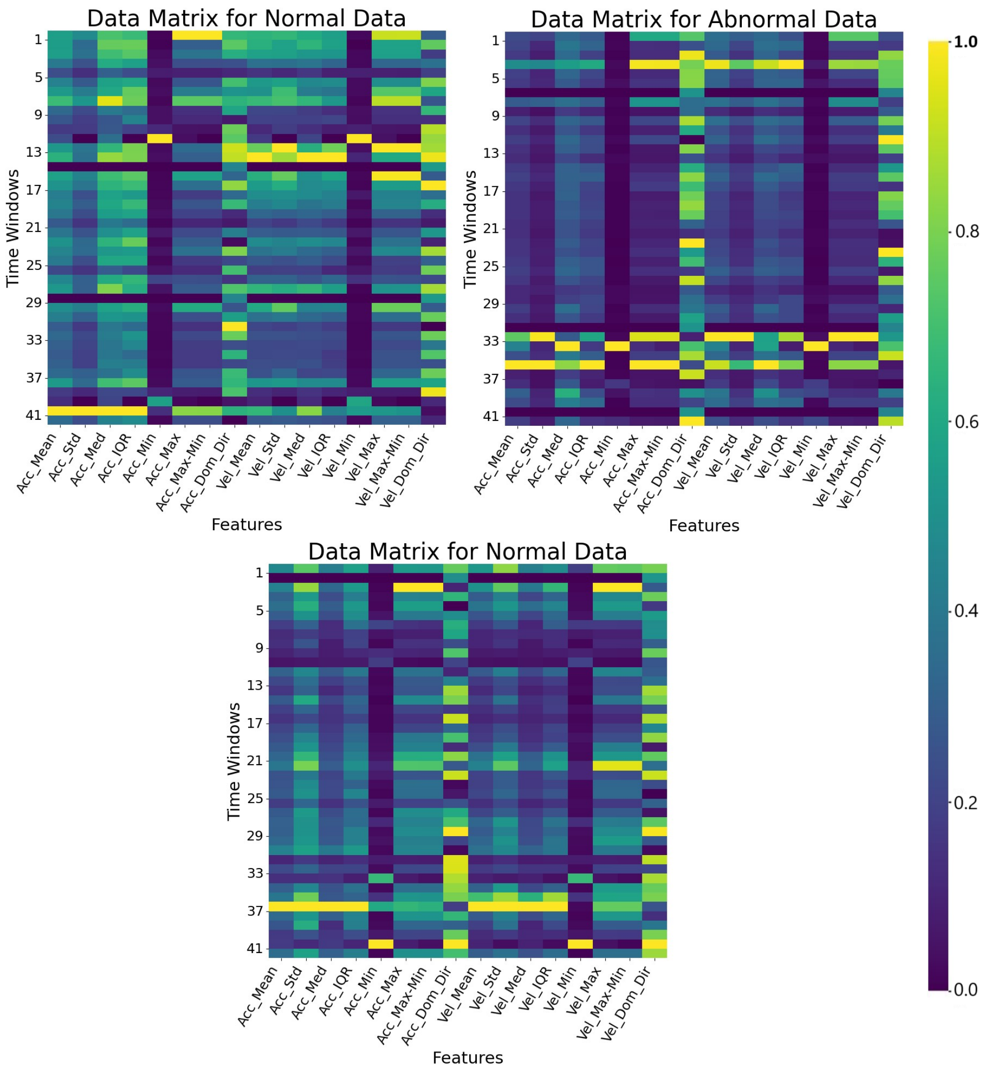

Figure A.1: Data Matrices for Subject D with an 80-second time sample

Appendix B

Evaluation Metrics for CNN+LSTM Model

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ELU	81.63	88.29	72.93	79.88
ReLU	83.98	90.73	75.69	82.53
Tanh	81.21	88.18	72.1	79.33
CNN-relu, Dense-elu	78.73	87.68	66.85	75.86
CNN-relu, Dense-tanh	81.63	91.04	70.17	79.25
CNN-elu, Dense-relu	80.8	86.79	72.65	79.1
CNN-elu, Dense-tanh	88.12	88.98	87.02	87.99
CNN-tanh, Dense-relu	88.95	88.95	88.95	88.95
CNN-tanh, Dense-elu	85.22	79.04	95.86	86.64

Table B.1: Evaluation Metrics for CNN+LSTM Model with Minority Class Oversampling

Activation Function	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ELU	83.51	77.43	94.59	85.16
ReLU	85.4	84.66	86.49	85.56
Tanh	85.13	78.51	96.76	86.68
CNN-relu, Dense-elu	84.59	78.07	96.22	86.2
CNN-relu, Dense-tanh	82.7	88.54	75.14	82.29
CNN-elu, Dense-relu	80.54	74.67	92.43	82.61
CNN-elu, Dense-tanh	85.41	81.64	91.35	86.22
CNN-tanh, Dense-relu	87.3	83.82	92.43	87.91
CNN-tanh, Dense-elu	881	84.06	94.05	88.76

Table B.2: Evaluation Metrics for CNN+LSTM Model with Majority Class Undersampling

Appendix C

Zones for Collective Anomaly Detection

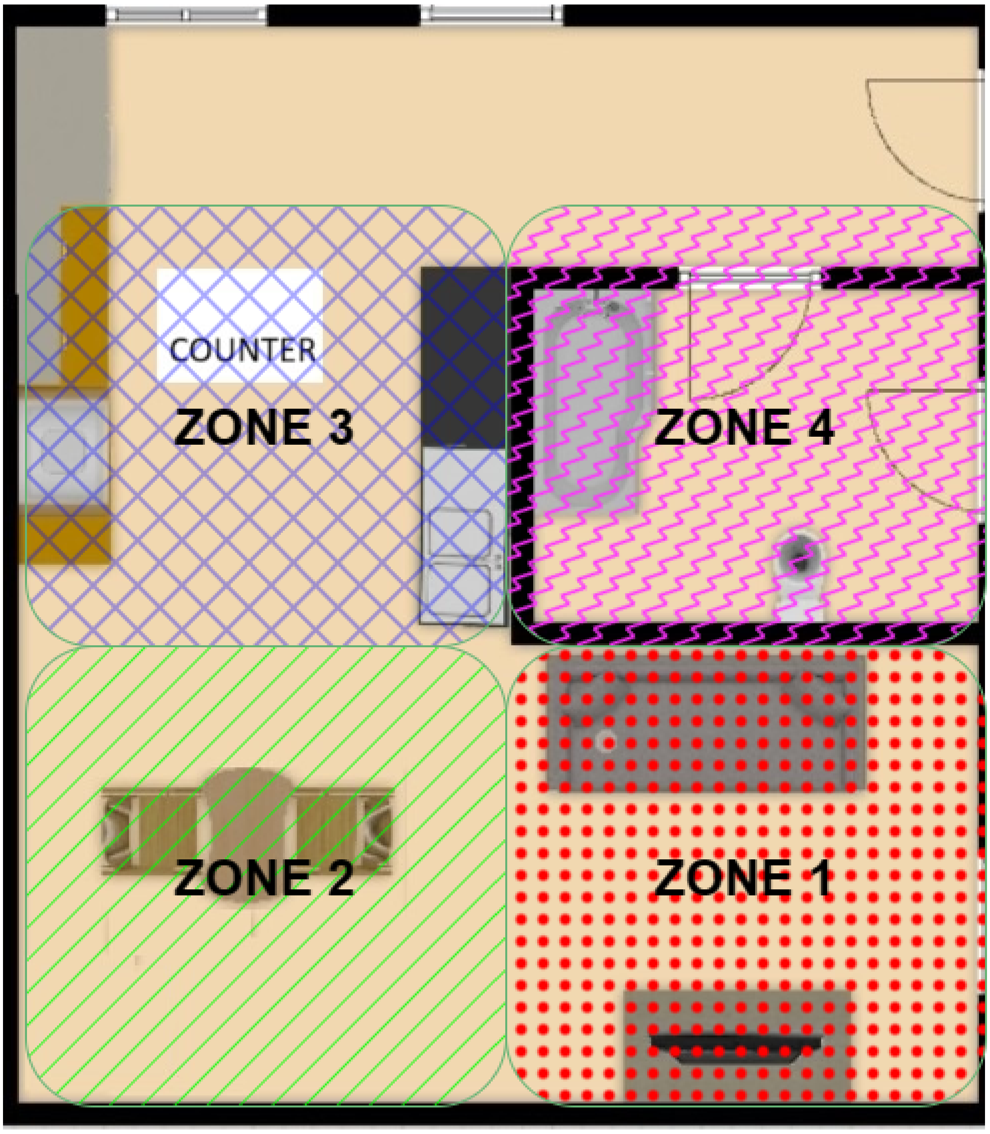

Figure C.1: Location of Zones in the Room for Collective Anomaly Detection

Appendix D

Participants' Information Sheet

Participant Information Sheet

Project title:	**Facilitating health and wellbeing by developing systems for early recognition of urinary tract infections - Feather**
Principal investigator (PI):	Kia Nazarpour
Researcher(s):	Lynda Webb; Saber Mirzaee, Emilyann Nault, Bhavith Manapoty, Jeong Younwoo, Aidan McConnell-Trevillion
PI contact details:	kianoush.nazarpour@ed.ac.uk

This study is certified according to the Informatics Research Ethics Process, RT number **671984**. Please take time to read the following information carefully. You should keep this page for your records.

Who are the researchers? The research team are members of the Feather Project from The University of Edinburgh, Heriot-Watt University and research partners. Kia Nazarpour is Principal Investigator. Nigel Goddard (UoE), Steve Leung (NHS Lothian), Lynne Baillie (HW) and Mauro Dragone (HW) are Co-Investigators. Saber Mirzaee Bafti, Lynda Webb, and Emilyann Nault who are researchers in the team. They may be accompanied by PhD and BSc students, Aidan McConnell-Trevillion, Bhavith Manapoty and Younwoo Jeong, while conducting this project.

What is the purpose of the study? The objective of this study is to explore how non-invasive sensing technologies, for example wearables and RADAR, can register movement patterns during the activities of daily living in a simulated home environment.

Do I have to take part? No – participation in this study is entirely up to you. You may decide to stop being a part of the research study at any time without explanation. You have the right to ask that any data you have supplied to that point be withdrawn or destroyed. You have the right to refuse to answer or respond to any question that is asked of you. You have the right to have your questions about the procedures answered (unless answering these questions would interfere with the study's outcome). If you have any questions as a result of reading this information sheet, you should ask the researcher before the study begins. You will have the option of taking part in the longer or shorter experiments.

What will happen if I decide to take part? The study will be conducted at the National Robotarium in the Laboratory of Robotic Assistive Living – LARA, which is an accessible home, based on the "Concept Blackwood House".

In this experiment, participants will be monitored over a three-hour period. The participants will engage in various everyday activities such as reading, eating, drinking, and more. During some periods they will be asked to simulate the experience of pain and the impact this pain may have on how you sit/move/walk.

The extent of this simulation is entirely under your control. Researchers will monitor your actions from a separate room, and our sensing technology including, for example a contactless radar device, a camera and a wrist-worn wearable device will track and record your movements throughout the experiment.

At the start of the experiment, you will receive instructions about the activities and their duration. You will be provided with all the equipment for these activities which include reading a magazine, watching TV, paying video games on tablets, completing a jigsaw puzzle. Making and consuming tea/coffee, preparing and eating a sandwich, and visits to an adjacent room/facility within the flat, where you will be requested to wait for a set amount of time.

Time Commitment The experiment will not take more than three hours and 15 minutes including the introduction and breaks.

Are there any risks associated with taking part? There are no risks associated with participation.

What will happen to the results of this study? The results of this study will be used to evaluate and iterate our sensing technology and inform the development of data analysis methods to detect changes in people's movement and behavioural patterns. The results may be summarised in published articles, reports and presentations. Data may also be used for future research. Raw data will be anonymised and archived on a public data repository as per the requirement of the funding agency.

Data protection and confidentiality Your movement records from our movement tracking device and the wearable, along with videos from the camera will be electronically stored on a secure hard drive. Your data will be processed in accordance with Data Protection Act (2018). All information collected about you will be kept strictly confidential. Your data will only be viewed by the research team, as described above. All electronic data will be stored on a password-protected encrypted computer, at The University of Edinburgh's School of Informatics' secure file servers and all paper records will be stored in a locked filing cabinet in the Principal Investigators office. Your consent information will be kept separately from your responses to minimise risk. The data will be kept as long as is required by Government Regulation, typically 4 years.

What are my data protection rights? The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance Data Protection

Law. You also have other rights including rights of correction, erasure and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk. Questions, comments and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk.

Who can I contact? Kia Nazarpour will be glad to answer your questions about this study at any time. You may contact him at kianoush.nazarpour@ed.ac.uk. If you want to find out about the final results of this study, you can contact Dr Nazarpour directly. If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk. When you contact, please provide the study title and detail the nature of your complaint.

Signature

By signing below, you are agreeing that:

- you have read and understood the Participant Information Sheet,
- questions about your participation in this study have been answered satisfactorily,
- you are aware of the potential risks (if any), and
- you are taking part in this research study voluntarily (without coercion).

Participant's Name (Printed)*: ________________
Participant's signature*: ________________
Date: ________________

**Participants wishing to preserve some degree of anonymity may use their initials.*

Appendix E

Participants' Consent Form

Participant Consent Form

Project title:	**Facilitating health and wellbeing by developing systems for early recognition of urinary tract infections - Feather**
Principal investigator (PI):	Kia Nazarpour
Researcher(s):	Lynda Webb; Saber Mirzaee, Emilyann Nault, Bhavith Manapoty, Jeong Younwoo, Aidan McConnell-Trevillion
PI contact details:	kianoush.nazarpour@ed.ac.uk

By participating in the study, you agree that:

- I have read and understood the participant information sheets (PIS) and I have had the opportunity to ask questions, and that any questions I had were answered to my satisfaction.
- My participation is voluntary, and that I can withdraw at any time without giving a reason. Withdrawing will not affect any of my rights. In addition, should I not wish to answer any particular question or questions, I am free to decline.
- I consent to my anonymised data being used in academic publications and presentations.
- I understand that my anonymised data will be stored for the duration outlined in the Participant Information Sheet.

Please tick yes or no for each of these statements.

		Yes	No
1.	I agree to being video recorded.	☐	☐
2.	I allow my data to be used in future ethically approved research.	☐	☐
3.	I understand that my responses will be kept strictly confidential. I give permission for members of the research team to have access to my anonymised data and responses. I understand that my name will not be linked with the research materials, and I will not be identified or identifiable in the report or reports that result from the research.	☐	☐
4.	I agree to take part in this study.	☐	☐

Name of Participant	Date	Signature
Name of person taking consent	Date	Signature
To be signed and dated in presence of the participant		
Name of Chief Investigator	Date	Signature

Once this has been signed by all parties the participant can receive a copy of the signed and dated consent form, participant information sheets and any other written information provided to the participants. A copy of the signed and dated consent form will be kept with the project's main documents in a secure location.